AD-A237 266

Technical Report 896

Optimization of Simulation-Based Training Systems: Model Description, Implementation, and Evaluation

Paul J. Sticha and H. Ric Blacksten
Human Resources Research Organization

Dennis M. Buede
Decision Logistics, Inc.

Michael J. Singer, Elizabeth L. Gilligan, Randall J. Mumaw, and John E. Morrison
U.S. Army Research Institute

A-1

June 1990

United States Army Research Institute for the Behavioral and Social Sciences

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS --
2a. SECURITY CLASSIFICATION AUTHORITY -- 2b. DECLASSIFICATION / DOWNGRADING SCHEDULE --	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
4. PERFORMING ORGANIZATION REPORT NUMBER(S) FR-PRD-88-26	5. MONITORING ORGANIZATION REPORT NUMBER(S) ARI Technical Report 896
6a. NAME OF PERFORMING ORGANIZATION Human Resources Research Organization 6b. OFFICE SYMBOL (If applicable) --	7a. NAME OF MONITORING ORGANIZATION ARI Field Unit at Orlando
6c. ADDRESS (City, State, and ZIP Code) 1100 South Washington Street Alexandria, VA 22314-4490	7b. ADDRESS (City, State, and ZIP Code) Attn: PM TRADE/ARI 12350 Research Parkway Orlando, FL 32826-2376
8a. NAME OF FUNDING / SPONSORING ORGANIZATION US Army Research Institute 8b. OFFICE SYMBOL (If applicable) PERI-IF	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER MDA 903-85-C-0169
8c. ADDRESS (City, State, and ZIP Code) 5001 Eisenhower Avenue Alexandria, VA 22333-5600	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 63007A PROJECT NO. 795 TASK NO. 345 WORK UNIT ACCESSION NO. C1

11. TITLE (Include Security Classification) Optimization of Simulation-Based Training Systems: Model Description, Implementation, and Evaluation

12. PERSONAL AUTHOR(S) Sticha, Paul J.; Blacksten, H. Ric (HumRRO) Buede, Dennis M. (Dec.Logis.) Singer, Michael J. (ARI) Gilligan, Elizabeth L.; Mumaw, Randall J.; Morrison, John E.(HumRRO)

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM 86/07 TO 88/10	1990, June	377

16. SUPPLEMENTARY NOTATION
--

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Training Resource Allocation; Training System Model; Training System Optimization; Medium Selection; Instructional Features; Fidelity

19. ABSTRACT (Continue on reverse if necessary and identify by block number)
A model for the optimization of simulation-based training systems was developed using a systematic, top-down design procedure. The model consists of five tools that address the following problems: a) determining which tasks should be trained by part-mission devices, full-mission simulators, or actual equipment; b) specifying instructional features needed to train a set of tasks efficiently; c) specifying the levels of fidelity that should be provided along several fidelity dimensions in order to meet task training requirements and satisfy cost limits; d) determining the family of training devices that can train all required tasks at minimum cost; and e) determining the optimal allocation of training time to training devices, given constraints on device use. The tools share common data on task requirements, training device features, and costs.

A prototype decision support system was developed, and a formative evaluation was conducted The model was demonstrated on Army rotary-wing aviation tasks, and specifications for application to armor maintenance were developed. The report describes the model using(OVER)

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Michael J. Singer	22b. TELEPHONE (Include Area Code) 407/380-8173 22c. OFFICE SYMBOL PERI-IF

DD Form 1473, JUN 86 Previous editions are obsolete. SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

ARI Technical Report 896

19. ABSTRACT (Continued)

the IDEFO (Integrated Computer-Aided Manufacturing Definition) system modeling language. In addition, the model is described for an example problem from a user's perspective. The example problem illustrates the way that the OSBATS model organizes the decision process, the kinds of analyses that can be performed, and the kinds of output that are produced. It was concluded that the prototype provides an adequate interactive environment in which the developer can perform several kinds of tradeoff analyses. The OSBATS software includes the data necessary to use the model for certain problems in Army rotary-wing aviation.

U.S. ARMY RESEARCH INSTITUTE
FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency Under the Jurisdiction of the Deputy Chief of Staff for Personnel

EDGAR M. JOHNSON
Technical Director

JON W. BLADES
COL, IN
Commanding

Research accomplished under contract for the Department of the Army

Human Resources Research Organization

Technical review by

Larry L. Meliza
Ray S. Perez

NOTICES

DISTRIBUTION: Primary distribution of this report has been made by ARI. Please address correspondence concerning distribution of reports to: U.S. Army Research Institute for the Behavioral and Social Sciences, ATTN: PERI-POX, 5001 Eisenhower Ave., Alexandria, Virginia 22333-5600.

FINAL DISPOSITION: This report may be destroyed when it is no longer needed. Please do not return it to the U.S. Army Research Institute for the Behavioral and Social Sciences.

NOTE: The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

Technical Report 896

Optimization of Simulation-Based Training Systems: Model Description, Implementation, and Evaluation

Paul J. Sticha and H. Ric Blacksten
Human Resources Research Organization

Dennis M. Buede
Decision Logistics, Inc,

Michael J. Singer, Elizabeth L. Gilligan, Randall J. Mumaw, and John E. Morrison
Human Resources Research Organization

PM TRADE Field Office at Orlando
Stephen L. Goldberg, Chief

Training Research Laboratory
Jack H. Hiller, Director

U.S. Army Research Institute for the Behavioral and Social Sciences
5001 Eisenhower Avenue, Alexandria, Virginia 22333-5600

Office, Deputy Chief of Staff for Personnel
Department of the Army

June 1990

Army Project Number
2Q263007A795

Training Simulation

FOREWORD

The cost of training devices and simulators has exceeded, in some cases, the cost of the operational equipment that they service. The capabilities for simulating reality are simultaneously increasing on an annual basis. The problem confronted by the military is to determine exactly how much simulation is sufficient for the stated learning objectives. Behavioral and analytical techniques that can quickly and easily project or predict how much simulation and training is required are lacking. At the same time information on the cost-effective use of training equipment within courses of instruction is sparse. The development of models, databases, and techniques addressing these problems provides the first steps toward providing integrated behavioral and engineering decisions in designing, fielding, and using advanced training technology. The potential effect on the Army is to reduce the cost of training equipment while increasing the equipment's instructional effectiveness.

In response to these concerns and problems, the Army Research Institute for the Behavioral and Social Sciences (ARI) and the Project Manager for Training Devices (PM TRADE) have joined efforts (MOU of Technical Coordination, May 1983; MOU Establishing the ARI Orlando Field Unit, March 1985; Expanded MOU, July 1986). Task number 3.4.5, Advanced Technology for the Design of Training Devices and Simulators, provided the impetus for the work reported in this document. PM TRADE has maintained partnership in all aspects of the development of the models, databases, and analytical techniques. The final prototype software was delivered to ARI and PM TRADE in December 1988, and has been disseminated to interested parties at Fort Rucker, the Army Training Support Command, and the Systems Training Directorate at the Training and Doctrine Command. The prototype has also been provided at their request to the Naval Training Systems Center Human Factors Research Group, the Air Force Aeronautical Systems Division, the Air Force Human Research Laboratory at Williams AFB, and National Aeronautics and Space Administration Ames Research Center. The models and techniques developed in this effort are expected to provide the basis for useful aids supporting the integration of behavioral and engineering data, knowledge, and expertise in training equipment design in the future.

EDGAR M. JOHNSON
Technical Director

ACKNOWLEDGMENTS

The authors would like to acknowledge the support they received in completing this project and preparing the reports. Dr. Halim Ozkaptan, the technical monitor for the project, provided sage counsel and guidance that set the direction for this effort. Mr. William Goodrick provided an ongoing link between the model developers and the user community at PM TRADE. Other engineers from PM TRADE evaluated the OSBATS software at various stages in its development. Many of their suggestions have been incorporated into the software.

Many members of the contractor research staff have also assisted in this project. Dr. C. Mazie Knerr provided management support at several critical points in the project and also worked on the development of the instructional-feature selection rules. Mr. Leon Elder and Mr. Peter DeYoe assisted in the preparation of model documentation. Ms. Marty Carson was responsible for document processing and preparation.

We are pleased with the working relationship we were able to develop and maintain with our subcontractors. The quality of the prototype software is due primarily to the efforts of Mr. James A. Stock from PAR Government Systems Corporation. Dr. Kenneth D. Cross and Mr. Carl Bierbaum provided the knowledge that was incorporated into the fidelity rule base. They were also responsible for developing the data used in the example problem.

OPTIMIZATION OF SIMULATION-BASED TRAINING SYSTEMS: MODEL DESCRIPTION, IMPLEMENTATION, AND EVALUATION

EXECUTIVE SUMMARY

Requirement:

The goal of this project is to develop and demonstrate methods that help training-device designers perform tradeoff analyses required for training-device design. These methods should allow the designer to determine the training-device alternatives that meet training requirements at a minimum cost or provide the maximum training effectiveness at a given cost. The methods should be usable and apply to the concept-formulation phase of the training-device development process.

Procedure:

A model for the optimization of simulation-based training systems (OSBATS) was developed using a systematic, top-down design procedure. The model consists of five tools that address the following problems: (a) determining which tasks should be trained by part-mission or full-mission simulators, and which should be trained on actual equipment; (b) specifying which instructional features are needed to train a set of tasks efficiently; (c) specifying the levels at which fidelity should be provided along several fidelity dimensions in order to meet task training requirements and satisfy a training-device cost limit; (d) determining the group of training devices that can train all required tasks at the minimum cost; and (e) determining the optimal allocation of training time to training devices, given constraints on device use. The tools share common data on task requirements, training device features, and costs. A prototype Decision Support System (DSS) implementing the OSBATS model was developed and formatively evaluated. The model was demonstrated on Army rotary-wing aviation tasks, and specifications for application to armor maintenance were developed.

Findings:

The OSBATS model is described as a system model using the IDEFO (Integrated Computer-Aided Manufacturing Definition) system modeling language. The IDEFO model provides a top-down analysis of model components and relationships. In addition, the model

activities are described for an example problem from a user's perspective. The example problem illustrates the way that the OSBATS model organizes the decision process, the kinds of analyses that can be performed, and the kinds of output that are produced. Finally, the results of a formative evaluation and an analysis of the activities required to apply the OSBATS model to the armor maintenance domain are described.

Use of Findings:

The OSBATS model may be used by engineers responsible for the development of a training-device concept to perform tradeoff analyses required to support the selection of the best technical approach to a training-device requirement. The prototype DSS provides an interactive environment in which the engineer may perform several kinds of tradeoff analyses. The OSBATS software includes the data necessary to use the model for certain limited problems in Army rotary-wing aviation.

OPTIMIZATION OF SIMULATION-BASED TRAINING SYSTEMS: MODEL DESCRIPTION, IMPLEMENTATION, AND EVALUATION

CONTENTS

LIST OF TABLES

LIST OF FIGURES

OPTIMIZATION OF SIMULATION-BASED TRAINING SYSTEMS: MODEL DESCRIPTION, IMPLEMENTATION, AND EVALUATION

INTRODUCTION

This report describes a project to develop, implement, and evaluate a prototype decision support system (DSS) that helps training-device designers provide for the Optimization of Simulation-Based Training Systems (OSBATS). The goal of the OSBATS model is to specify training concepts, training-device designs, and allocations of time to different training device alternatives that either minimize the training device costs required to meet training objectives or maximize the training device effectiveness obtained at a specified cost.

The first year of this effort produced the conceptual framework for the OSBATS model (Sticha, Blacksten, Buede, & Cross, 1986), plans for implementation and evaluation (Young, Luster, Stock, Mumaw, & Sticha, 1986), and a review of the research literature relevant to the goals of the OSBATS model (Sticha, Blacksten, Knerr, Morrison, & Cross, 1986). In the second year, our effort concentrated on revising and enhancing the model, developing prototype decision-support software implementing the OSBATS model (Elder, Gilligan, & Sticha, 1988), and conducting a preliminary evaluation of software modules (Sticha, Singer, Blacksten, Mumaw, & Buede, 1987). Development and evaluation efforts have continued during the final year of the project. This report describes the prototype OSBATS model, its implementation, and the results of the formative evaluation.

Goals of the OSBATS Model

A training system consists of several major components, including instructors, students, training equipment, instructional strategies, and tasks to be trained. The major concern of the OSBATS model is with the design and use of training equipment, particularly equipment that simulates the operation of part or all of the weapon system. The overall goal of the OSBATS model is to determine the kinds of training devices that can meet training requirements at the minimum cost. Training requirements are defined by the specific tasks that the student must be able to perform after training. Within this general framework, we have restricted the scope of the OSBATS model in several respects.

1. We are concerned principally with tasks that can be trained by training devices or simulators. We are not concerned with tasks that can be trained adequately in a classroom.

2. We are concerned with training devices that interact with the student dynamically in a manner that is analogous to the interactions that occur with actual equipment. This restriction implies that we are not interested in such training media as movies, videotapes, static representations of actual equipment, and other training aids that serve primarily to enhance classroom training. We are concerned with computer-based training (CBT) to the extent that the training involves a dynamic representation of the tasks

being trained, rather than a static presentation of information. There is a fine line between part-task training, which abstracts critical components of a task and provides training for these components in an environment that may be only slightly similar to the actual environment, and CBT that merely presents information to the student. We have not attempted to define precisely the boundary between these two types of training.

3. We are concerned with institutional training rather than unit training. Unit training involves complexities that were judged to be too difficult to handle in an initial implementation. However, it should be possible to generalize the model procedures to apply to unit training at a later date.

We have decomposed this overall goal into three sub-goals, and have developed a set of tools to meet these sub-goals.

The first sub-goal is to identify tasks that are good candidates for training using a training device. Tasks may be candidates for device-based training for several reasons. First, the use of a training device may provide training at a lower cost than comparable training on actual equipment. Second, a training device may be able to produce special environmental conditions that would be unsafe, expensive, or impossible to produce using actual equipment. Finally, a training device may be more efficient by allowing the student more repetitions of the tasks during the same training time than the actual equipment would, or by using appropriate instructional features. A second element of this sub-goal is to determine clusters of tasks that have similar training-device needs. The task clusters produced by this process form the requirements that are used as the basis for training-device design.

The second sub-goal is to design training devices with a level of sophistication and cost that is tailored to the requirements of the tasks for which they are designed. The major training device components considered in this problem either simulate the equipment and environment or provide instructional support to the training process. The simulation components may vary with respect to the fidelity with which they represent corresponding actual-equipment components. There are many simulation components, such as the device's visual system, motion system, and dynamic simulation system. The value of investing in different levels of fidelity for these components depends on the effectiveness of the components in reaching the training requirements. The device-design process must determine the minimum level of fidelity required by the tasks to be trained. The effectiveness of instructional features depends upon the characteristics of the tasks to be trained and the population of students. A training device should be designed with the instructional features that provide the greatest improvement in training efficiency given the tasks to be trained.

The third sub-goal is to determine the way to allocate training resources among training devices and actual equipment in order to minimize training cost. In some situations, it may be possible for a training device to provide cost-effective training on tasks other than those for which it was designed. In this case, the training device should be used for those tasks. In other situations, it may not be possible to provide the required fidelity to train a task at an acceptable cost. In this case, it may be optimal to design a simpler training-device that would replace only a portion of the training time on actual equipment. In its complete formulation, a procedure for allocating training among training devices and actual equipment must consider the constraints on the use of devices and actual equipment that come from budgetary limitations, space and equipment availability, and safety concerns.

Some of the complexity of training-system design is caused by interactions between the three sub-goals. That is, although there is a general logical progression through the sub-problems, later processes can provide feedback to earlier processes. For example, the resource allocation process that addresses the third sub-goal may indicate that a high-cost simulator design leads to a lower overall training cost than either a moderate- or a low-cost device. This feedback may lead the analyst to develop and evaluate other high-cost device designs. On the other hand, the resource allocation process might indicate that a low-cost training device can provide adequate training effectiveness for all but a small subset of the tasks. This result might prompt the analyst to design a new training-device specifically tailored to the tasks that could not be trained by the low-cost device.

The interactions between the sub-goals for the OSBATS model imply that a simple linear approach to the problem will not work in some cases. Because of the complexity of the sub-goal interactions, any model must be designed to be used iteratively. That is, the results of individual model components must provide input to later components and feedback to earlier components. The OSBATS model provides for iterative application of its component modules with greater precision at each application cycle. The sub-goal interactions also indicate the need for sensitivity analyses in which model assumptions are varied to ensure that the solution obtained is a global, rather than local, optimum.

Model Description

The OSBATS model was developed iteratively using a top-down, system-analytic approach. The overall problem of training-system optimization was decomposed into the sub-goals, which were, in turn, decomposed into individual modules. Five software tools were developed to address the three sub-goals. One tool, the Simulation Configuration Module, addresses the first sub-goal. Two tools, the Instructional Feature and Fidelity Optimization Modules, address the second sub-goal. Two tools, the Training Device Selection and Resource Allocation Modules, address the

third sub-goal. The function of the five tools is briefly described below, and is followed by an introductory presentation of the input data requirements, the model processes, and type of recommendations generated by the model.

1. The Simulation Configuration Module clusters tasks to be trained according to their need for training on a full-mission simulator (FMS), one or more part-mission simulators (PMSs), or actual equipment (AE).

2. The Instructional Feature Selection Module determines the relative priority with which instructional features should be included in a training device.

3. The Fidelity Optimization Module determines the relative priority of features that allow a training device to represent the actual equipment and operational environment.

4. The Training Device Selection Module selects a set of training devices that can be used to meet the training requirements for each task at the least cost.

5. The Resource Allocation Module determines the optimal allocation of training time to training devices and actual equipment to meet all training requirements, considering constraints on device procurement and use.

Input Data Requirements

All methods of training system design require a good front-end analysis. The OSBATS model is no exception to this rule, and requires information about training requirements, task characteristics, trainee population skills, candidate training-device instructional features, and fidelity dimensions. In addition, because the model is quantitative rather than qualitative, it requires numerical estimates for many of its parameters. We do not anticipate that the engineer using the OSBATS model will be the principal individual responsible for providing input data. Rather, we see two principal sources of data for the OSBATS model. First, information about the problem structure, general training-device characteristics, and inference rules will be resident in the model. The second class of data describes the specific training problem addressed by the OSBATS model. This information describes the tasks to be trained according to the parameters of the model components. We do not expect that the user will have the subject-matter expertise required to provide these data directly. Consequently, we envision that ultimately these data will be developed through a task analysis that supports the training-device design process. Certain inputs are required of the user, however. These inputs consist of the critical judgments that express general priorities in training-system design, and that limit the scope of the problem addressed by the OSBATS model.

Model Processes

The overall modeling framework is based on methods that define the training strategy that meets the training requirements at the minimum cost. This framework was originally described by Roscoe (1971) and has been extended by Povenmire and Roscoe (1973), Carter and Trollip (1980), Bickley (1980), Cronholm (1985), and our own work (Sticha, Blacksten, Buede, & Cross, 1986; Sticha, Singer, Blacksten, Mumaw, & Buede, 1987). In its simplest form, the method compares the ratio of effectiveness of two training alternatives to the ratio of cost of the options. For example, if a training program that employs one hour of training on a simulator saves 30 minutes of training on actual equipment, and the hour of simulator training costs as much as 20 minutes of training on actual equipment, then the simulator will meet the training requirement at a lower cost than actual equipment. This approach addresses the tradeoff between the increased training time that is usually required by the use of a simulator and the decreased cost of that time.

The goal is to develop training device designs that have the fidelity and instructional features required to meet the training requirements for the tasks while avoiding extraneous or inefficient features. We have applied a general design methodology to the analysis for training-device design. This methodology addresses problems in which there are many alternatives formed by the factorial combination of several dimensions. We have developed two applications of this methodology. The first application addresses the instructional features that should be included in the training device; the second application addresses the fidelity features that should be included.

Model Recommendations

The concept of operation for the OSBATS model is based on the iterative use of the five model tools to make recommendations regarding the definition of task clusters, the design of training devices, and the allocation of training resources among selected training devices. Both the subset of tools that are used and the order in which they are used may vary depending on the requirements of the problem and the preferences of the user. Although the tools may be used in a variety of orders, the most natural order is to address the three sub-problems in the order that they were originally enumerated. Later in the report, an example application of the tools in that order is presented that illustrates the kinds of recommendations that may be made by the OSBATS model.

An overview of the OSBATS model is presented in the second section of the report. A description of the model tools, in terms of the processes used by the tools and the generated output, is provided first. That is followed by a summary of the rulebases used to identify features, and a summary of data

requirements. The final portion of the section discusses model the model assumptions.

Model Implementation

The second major activity of the project was the development of software to embody the OSBATS model. The implementation of the model is embodied by the complete model description provided in the third section of the report. The IDEFO methodology is used to provide a detailed representation of the model as implemented. The implemented software provides a prototype DSS that can interact with an analyst about a simplified training system in Army Aviation. The demonstration example is based on the AH-1 Airman Qualification Course (AQC), and is described in the fifth section of the report.

The OSBATS software runs on an IBM PC/AT, Zenith 248 or compatible with 640K of memory, and a high density storage device. In addition, the following features are required: (a) an enhanced graphics adapter (EGA) and color EGA monitor, (b) 80287 numeric coprocessor, and (c) a Microsoft-compatible mouse.

Individual modules were delivered for evaluation as they were developed. The evaluation results were used to guide the development of later modules. The delivery schedule is shown below:

December 1986 - Instructional Feature Module
February 1987 - Fidelity Optimization Module
March 1987 - Simulation Configuration Module
April 1987 - Training-Device Selection Module
June 1987 - Resource Allocation Module
July 1987 - Integrated OSBATS Prototype (version 1.0)
March 1988 - Revised OSBATS Prototype
October 1988 - Final OSBATS Prototype (version 1.1)

The final version of the software incorporates changes suggested by the results of the evaluation (see below) of both the integrated and revised versions.

The software implements all of the modeling capabilities described in the IDEFO presented model design. The software allows the user to define task clusters, develop candidate training-device designs for individual task clusters, and evaluate the cost of providing training using these designs in various combination with or without actual equipment. The software does not provide the capability to enter or modify the data required by the model.

Model Evaluation

We conducted evaluations of the model and software at several points in the development process. A detailed presentation of the evaluation efforts, analyses, and conclusions

is presented in the fourth section of the report. The following material provides an overview of the evaluation efforts, and how that information was used.

First, we conducted formative evaluations of the prototype OSBATS modules as they were delivered. The evaluation consisted of interviews with PM TRADE engineers as a newly delivered module was demonstrated. The primary goal of the initial evaluation was to provide feedback to the development process regarding the perceived utility of the model and the adequacy of the user interface.

Our second evaluation involved a verification of the methods used by the software against the description of the model. When discrepancies were found, they were examined to find their source. In some cases, our examination of the discrepancies uncovered errors in the model concepts or methods, which were corrected. Discrepancies that were caused by an error in the implementation of the model were communicated to the programming staff, so that they could be corrected. To our knowledge, all of these errors have been corrected.

A third form of evaluation involved a detailed demonstration of the software and structured interview with potential model users from PM TRADE. The interviews addressed the presentation of data and results, the clarity of the model's results, the validity of the model's approach, the availability of data required by the model, and the degree to which all relevant information was included in the model. Some of the results of this evaluation have been used to guide model revisions. Others were used to guide our recommendations for future revisions.

A fourth form of evaluation involved performing sensitivity analyses on critical model parameters. In these analysis, the values of input data were varied, and the changes in the overall results were examined. The sensitivity analyses helped us to pinpoint the input variables that are most critical to the model results. These critical variables are the ones that need to be known with the greatest accuracy, and also are the best candidates for future research.

Our final evaluation involved an analysis of the requirements for the application of the OSBATS model to a completely different domain, namely Armor Turret Maintenance. This analysis investigated the extent to which data variables and modeling constructs would need to be changed to apply OSBATS to this new domain.

Organization of this Report

The next or second section in the report contains a general overview of the model components. The section describes the data they require, the methods they use, and the outputs they produce.

In the following section, we present a description of the IDEF0 (Integrated Computer-Aided Manufacturing Definition) system and a formal system description of the implemented model using the IDEF0 system methods. The IDEF0 model provides a top-down analysis of model components and their relationships.

The fourth section describes the evaluation activities that have been conducted. The evaluations address the accuracy of the model, its utility, and the quality of the user interface.

The next section of this report presents an example that illustrates the application of the OSBATS model to a specific problem related to the AH-1 AQC. The example problem illustrates the kinds of outputs that are produced by the model software and the types of interactions that are possible. This example should orient the reader to the decisions that are addressed by the OSBATS model, and the approaches that the model uses to aid these decisions.

The following section describes the application of the OSBATS model to Armor Turret Maintenance training. This evaluation is a conceptual analysis that addresses the generality of the OSBATS model to a variety of training domains.

The seventh and final section of the report discusses several of the issues that are relevant to model development and use. It also summarizes the major accomplishments of this project and describes additional model-development needs.

MODEL OVERVIEW

This section gives an overview of the OSBATS model to prepare the reader for the more detailed description that is given in the following section. The descriptions in these sections focus on how the model works. Thus, we will describe the input data requirements and model processes in some detail.

The section begins with an overview of the model description, elaborating on the description presented in the introduction. Then a summary of each modeling tool or module, describing the input data, model process, and model output is presented. We then describe the two rule bases that support the operation of the models. Then we summarize the input data requirements for all of the modules. Finally, some of the simplifying assumptions that have been used in the OSBATS model are summarized.

Model Description

As stated in the introduction, the OSBATS model was developed iteratively using a top-down, system-analytic approach. The overall problem of training-system optimization lead to the development of five software tools. The top-down structuring methods were extended to the development of each tool. First, general procedures and analysis strategies were developed. The general procedures specify the kinds of variables that are relevant to the tool and how they are combined. For example, the general procedures specify whether a tool considers life cycle cost or development cost only, what factors are considered in the determination of effectiveness, and whether the recommendation of the tool is based on an effectiveness/cost ratio or another mathematical optimization procedure. Then specific procedures were obtained by examining the current knowledge and supplementing this knowledge, where necessary, with reasonable conjectures. The specific procedures provide the detailed methods used to calculate relevant cost and effectiveness measures. The resulting tools are general in that they apply to a wide variety of training systems, simulators, and other training devices. The input data used by the model contain information that is specific to our initial application to aviation training.

To give an overview of the OSBATS model, we briefly describe the input data required by the model, the processes used by the model, and the recommendations produced by the model.

Input Data Requirements

All methods of training system design require a good front-end analysis. The OSBATS model is no exception to this rule, and requires information about training requirements, task characteristics, trainee population skills, candidate training-device instructional features, and fidelity dimensions. In addition, because the model is quantitative rather than qualitative, it requires numerical estimates for many of its parameters.

The user of the OSBATS model should not be the principal individual responsible for providing input data. Information about the problem structure, general training-device characteristics, and associated inference rules is resident in the model. This information must be updated periodically as the domain of the model expands, and as new research results are developed. The updating will require data entry of some kind, possibly performed by the user. Task data is input to describe the specific training problem to be addressed by the OSBATS model. This information describes the tasks to be trained according to the parameters required by the model components. These data are traditionally developed through a task analysis that supports the training-device design process.

There are inputs required of the user, consisting of the critical judgments that express general priorities in training-system design, and that limit the scope of the problem addressed by the OSBATS model. Direct access to relevant data will eventually have to be provided through a database management system. When the database management system is integrated with the OSBATS model, the user should be able to examine and modify much of the data used by the model.

Model Processes

The overall modeling framework is based on methods that define the training strategy that meets the training requirements at the minimum cost. This framework was originally described by Roscoe (1971) and has been extended by Povenmire and Roscoe (1973), Carter and Trollip (1980), Bickley (1980), Cronholm (1985), and our own work (Sticha, Blacksten, Buede, & Cross, 1986; Sticha, Singer, Blacksten, Mumaw, & Buede, 1987). In its simplest form, the method compares the ratio of effectiveness of two training alternatives to the ratio of cost of the options. For example, if a training program that employs one hour of training on a simulator saves 30 minutes of training on actual equipment, and the hour of simulator training costs as much as 20 minutes of training on actual equipment, then the simulator will meet the training requirement at a lower cost than actual equipment. This approach addresses the tradeoff between the increased training time that is usually required by the use of a simulator and the decreased cost of that time.

This simple formulation of the cost-effectiveness of training may be used to provide recommendations regarding the optimal mix of simulator and actual-equipment training when the effectiveness of training on the simulator is expressed as a function of the amount of simulator training. In general, the effectiveness of simulator training is a negatively accelerated function of the amount of training on the simulator. The optimal mix of simulator and actual-equipment training involves training on the simulator until the marginal cost savings from reduced actual-equipment usage equals the marginal cost of the simulator training. Cronholm (1985) generalized this finding by splitting the transfer function into three components representing learning on the simulator,

transfer to the actual equipment, and learning on the actual equipment, respectively.

We have extended the basic modeling framework in two ways in the OSBATS model. The two extensions are implemented in the Training Device Selection and Resource Allocation modules of the OSBATS model. Both extensions make the same assumptions about learning and transfer processes. The Training Device Selection Module makes simplifying assumptions about training cost so that it can provide an interactive environment for addressing training-device selection alternatives. The Resource Allocation Module relaxes some of the assumptions to allocate training resources to training devices considering both discrete purchase costs and device use constraints.

This basic framework, as we have extended it, provides the overview to the decision process. We have combined these methods, which address the third subgoal, with general methods addressing the other two subgoals. These two methods include a preliminary clustering method, implemented in the Simulation Configuration Module, and a training-device design method, implemented in the Instructional Feature Selection and Fidelity Optimization Modules.

Preliminary clustering method. The goal of the preliminary clustering method used by the Simulation Configuration Module is to review task requirements, simulation needs, and cost of simulation capability in order to define clusters of tasks that have similar simulation requirements. The method currently defines the following three classes of training devices: (a) a full-mission simulator (FMS) that simulates many or all of the subsystems of the actual equipment, (b) one or more part-mission simulators (PMSs) that simulate selected equipment subsystems, or (c) actual equipment.

This evaluation examines device-unique capabilities, such as training in unsafe situations, and cost savings to establish the value of training with some sort of training device. In addition, the task requirements for fidelity are used to estimate the development cost that would be required to achieve the required fidelity for each task. Using the assessed costs and benefits, the model sorts the tasks into three clusters: (a) those tasks that should be trained on actual equipment because the benefits of simulation do not justify the expense required to develop an effective training device, (b) those tasks for which training in a simulated environment is cost-effective and which have limited cue and response requirements so that they require only a PMS, and (c) those tasks for which training in a simulated environment is cost-effective, and which require an FMS because they require a high-fidelity representation of the environment on several dimensions.

The model makes its major recommendation regarding whether a task should be trained on actual equipment or on a training device by comparing the required development cost of the training device to the potential operating-cost savings brought about by its use.

If the operating-cost savings is sufficient to recover the development cost over the life cycle of the weapon system, a training device will be recommended. Otherwise, actual equipment will be recommended. The recommendations of the economic analysis are overridden, however, if a training device is required for safety considerations.

Training-device design methodology. The task clusters defined by the above procedure provide the requirements for individual training devices. The need at this point is to develop training device designs that have the fidelity and instructional features required to meet the training requirements for the tasks in a single cluster while avoiding extraneous or inefficient features. We have applied a general design methodology to the analysis for training-device design. This methodology addresses problems in which there are many alternatives formed by the factorial combination of several dimensions. We have developed two applications of this methodology. The first application addresses the instructional features that should be included in the training device; the second application addresses the fidelity features that should be included.

The OSBATS model views instructional features as elements of training devices that can improve training efficiency on individual tasks. That is, instructional features reduce the time or cost required to achieve a given performance level on a training device. They do not affect the ultimate level of actual-equipment performance that can be reached by using a training device. The number of tasks aided by each instructional feature forms the basis of an index of benefit for the feature. The analysis proceeds by comparing the benefit to the cost of incorporating each instructional feature into the training device. The analysis then orders the features according to the ratio of benefit to cost. This order specifies a collection of optimal features as a function of the total budget for instructional features. The appropriate budget for instructional features, given a total training-device budget, is determined in the Fidelity Optimization Model.

The same modeling framework is then used to address how much should be invested in the fidelity of the training device being designed. The model considers several dimensions of fidelity that describe task cue and response requirements. The task requirements on the fidelity dimensions are compared to the cost of meeting these requirements to determine the dimensions for which increased fidelity is justified by increased training effectiveness. The output of this model is a set of possible training-device configurations applicable to the task set, each of which offers the greatest effectiveness for its cost.

The model makes its selection based on the incremental benefit/cost ratio of the fidelity dimension levels. The costs are calculated from the fidelity levels, and represent development costs. The benefits are based on the number of task fidelity requirements that are met for each fidelity dimension level.

Model Recommendations

The OSBATS model operational concept is based on the iterative use of the five model tools to make recommendations regarding the definition of task clusters, the design of training devices, and the allocation of training resources among selected training devices. Both the subset of tools that are used and the order in which they are used may vary depending on the requirements of the problem and the preferences of the user. Although the tools may be used in a variety of orders, the most natural order is to address the three subproblems in the order that they were originally enumerated. An application of the tools in that order illustrates the kinds of recommendations that may be made by the OSBATS model.

The Simulation Configuration Module would be used first to examine the tasks to be trained to provides a preliminary recommendation for the use of either actual equipment or one or more training devices. The result of this analysis would be three clusters of tasks. Two of these clusters define tasks for which a full-mission simulator or part-mission training device should be designed.

The analyst would then use the task clusters defined by the Simulation Configuration Module as the basis for the application of the Instructional Feature Selection and Fidelity Optimization Modules. These two modules would be used to define candidate training system designs for each task cluster. The output of the two modules is a range of options that vary in cost. Thus, the overall results of the application of these modules would be a collection of training device designs specifying for each design the level of fidelity on each fidelity dimension and the collection of instructional features included in the design. The analyst would select several of these designs for further examination.

The Training Device Selection Module evaluates the training device designs produced in the previous process. The analyst would exercise this module several times using different combinations of training devices. For each combination, the module would determine the number of tasks that would be assigned to each training device, the number of hours each task would be assigned to each device to meet the training requirements at the lowest cost, and the optimal training cost given the particular combination of training devices. This model makes the simplifying assumptions that the hourly cost of a training device is fixed and that all devices are fully utilized. These assumptions allow the Training Device Selection Module to determine a solution in less than one minute.

When the analyst was relatively confident of the solution of the Training Device Selection Module, he or she would then investigate the solution using the Resource Allocation Module. It could be that the recommendations of the Training Device Selection Module would require the procurement of more training devices than would be feasible, or would provide some training on actual equipment for tasks in which such training violated safety

regulations. The Resource Allocation Module allows the analyst to impose constraints such as these on the training system and examine the resulting optimal solution. The Resource Allocation Module also relaxes the simplifying assumptions that were used by the Training Device Selection Module to estimate training device cost, leading to a more accurate cost function. As a result of its increased generality, the Resource Allocation Module takes several minutes to reach a solution, much longer than the Training Device Selection Module.

At many points in the analysis process the analyst has the option of returning to modules that were used previously to refine the analysis, change assumptions, or choose different solutions. For example, the analyst might change the definition of the task clusters based on the results of Training Device Selection Module, or may use those results to select different candidate device designs for evaluation.

Summary of Modeling Tools

This section presents a brief discussion of each of the tools, including descriptions of the input requirements, the outputs, and an outline of its method.

Simulation Configuration Module

The tool for simulation configuration examines task characteristics to determine both the need for and the cost-effectiveness of training on general classes of training devices as compared to training on actual equipment. Three classes of training devices are considered by the tool:

1. A Full-Mission Simulator (FMS), which simulates many or all of the subsystems of the actual equipment,

2. One or more Part-Mission Simulators (PMSs), which simulate selected equipment subsystems, or

3. Actual equipment.

This evaluation examines device-unique capabilities, such as training in unsafe situations and cost savings, to establish the value of training with some sort of training device. In addition, the task cue and response requirements, which are derived using set of rules that associate activities with requirements, are used to estimate the development cost required to achieve the necessary fidelity. Using the assessed costs and benefits, the model sorts the tasks into three clusters:

1. Those tasks that should be trained on actual equipment because the benefits of simulation do not justify the expense required to develop an effective training device.

2. Those tasks for which training in a simulated environment is cost-effective, and which have limited cue and response requirements so that they require only a PMS.

3. Those tasks for which training in a simulated environment is cost-effective, and which require an FMS, because they require a high-fidelity representation of the environment on several dimensions.

Input requirements. The primary inputs of the Simulation Configuration Module consist of ratings of tasks regarding safety concerns, necessary special performance conditions, and requirements for cue and response fidelity. The following types of inputs are required:

1. Task learning points. Numerical assessments of entry-level performance and the performance standard for each task.

2. Task simulation evaluation factors. Ratings regarding safety concerns, special performance conditions, and training effectiveness considerations. These factors are combined to produce an index of simulation indicators.

3. Task information processing requirements and activity data. These data are used by the fidelity rules to derive the task cue and response requirements, which, in turn are used to estimate the cost of developing a training device that can provide training to the performance standard. This cost measure provides the second major component of the model.

4. Task training hours and costs. The time that would be required to train each task on actual equipment to the performance standard. For tasks in which training on actual equipment is impossible, the training time is an estimate of the training time required on a hypothetical training device with near-perfect fidelity and without any instructional features.

5. Training-device costs. These measures describe the development, fixed operating, and variable operating costs of the training device templates and actual equipment. Also included in these data are estimates of maximum annual utilization and the length of the device life cycle.

6. Training-device cue and response capabilities and instructional features. These data describe the cue and response capabilities and instructional features present in currently developed or hypothetical training devices. One of these device descriptions is used as a baseline training device to estimate the operating-cost savings derived from simulation.

7. Fidelity dimension cost data. These data are used to estimate the development cost of candidate training devices from the cue and response requirements.

8. Fidelity Rules. These rules derive task cue and response requirements from task activity data and information processing requirements.

9. Training-system data. These data include user judgments of importance and criteria used by the tool to make recommendations.

Module output. This module recommends which tasks should be trained on a full-mission simulator, part-mission simulator, or actual equipment. The assignment of tasks to clusters is used by the Instructional Feature Selection and Fidelity Optimization Models.

Method. The module evaluates simulation options for each task by comparing the benefits of simulation due to improved safety or operating-cost savings to the cost required to develop task-specific effective simulation capability. For tasks in which the benefit of simulation is high, or for which simulation is required for safety concerns, a high-cost simulator may be justified. For tasks in which the benefit from simulation is moderate, a simulator would be justified only if it were less expensive to develop.

The benefits of simulation are assessed directly from ratings on safety concerns, special performance conditions, and training effectiveness factors. Operating-cost savings is determined by comparing the cost of training on actual equipment to the projected cost for an optimal mix of training on both actual equipment and a prototype training device. The cue and response requirements for each task are used to estimate the cost to develop a training device with the required capability for that task.

The model makes its major recommendation regarding whether a task should be trained on actual equipment or some kind of training device by comparing the required development cost to the potential operating cost savings. If the operating cost savings is sufficient to recover the development cost over the life cycle of the weapon system, a training device is recommended. Otherwise, actual equipment is recommended. The recommendations of the economic analysis is overridden, however, if a training device is required for safety considerations.

Instructional-Feature Selection Module

Instructional features are elements of training devices that can improve training efficiency on individual tasks. That is, instructional features reduce the time or cost required to achieve a given performance level on a training device. They do not affect the ultimate level of actual-equipment performance that can be reached using a training device. The analysis proceeds by deriving a benefit measure for each instructional feature, and comparing the benefit to the cost of incorporating each instructional feature into the training device. The number of tasks aided by each instructional feature forms the basis of an index of benefit for the feature. The cost of the feature is assessed directly. The analysis then orders the features according to ratio of benefit to cost. This order specifies a collection of optimal features as a function of the total budget for instructional features. The appropriate budget for instructional features, given a total training-device budget, is determined in the Fidelity Optimization Module.

Input requirements. Module inputs include information about both the tasks assigned for training on the training device and the candidate instructional features. The types of input requirements are:

1. Tasks cluster assigned to the training device. The list of tasks that will be used as the basis of the assessment of instructional-feature benefit could be based on the recommendations of the Simulation Configuration Module or the Training-Device Selection Module, or tasks could be selected by the user.

2. Task learning points and task information processing requirements. These data provide the information used by the instructional feature rules listed below.

3. Task training hours and costs. The time that would be required to train each task on actual equipment to the performance standard. For tasks in which training on actual equipment is impossible, the training time is an estimate of the training time required on a hypothetical training device with near-perfect fidelity and without any instructional features. The costs are normalized to produce task weights used in the analysis.

4. Instructional feature rules. These rules associate task characteristics with instructional features. For example, a rule might associate the tutorial feature with tasks that are in early phases of training (entry performance level is low) and are difficult.

5. Instructional-feature cost. This factor assesses the procurement cost associated with each instructional feature. The Instructional Feature Module requires only ratio-scale costs for the analysis. However, the actual range of costs is

required to incorporate the results of the Instructional Features Module into the Fidelity Optimization Module.

6. Instructional feature weight. This factor is a task-independent measure of instructional-feature value that estimates the probability that an instructional feature would be used. This information would be based on research such as that of Polzella (1983) and would be updated as new research results were obtained.

Module output. The major output of this module is the optimal ordering in which instructional features should be included in a training device as a function of the budget for instructional features. This list may serve as an input to the Fidelity Optimization Model, or it may be used directly to identify particularly useful instructional features for the selected task cluster.

Method. The analysis proceeds in three steps. In the first step, the user selects the candidate features that will be evaluated in the analysis and the cluster of tasks that will be used as the basis of the evaluation. The task cluster may come from the recommendations of the Simulation Configuration Module or the Training-Device Selection Module, or may be selected directly by the user. In the second step, the benefits of the instructional features are determined. The instructional-feature rules are applied to determine the features that are appropriate for each task. The appropriateness assessments are combined across tasks with the task and instructional-feature weights to produce a task-specific assessment of instructional-feature benefit. Thus, the benefit of an instructional feature is the weighted proportion of tasks for which the feature is appropriate. In the third step, the optimal list of instructional features is calculated. The benefit of each instructional feature is divided by its assessed cost, and the features are ordered according to the generated benefit-to-cost ratio. One or more user-selected, optimally ordered packages of instructional features may serve as input to the Fidelity Optimization Module.

Fidelity Optimization Module

This module addresses the problem of how much should be invested in the sophistication of a training device that allows it to represent the task-performance environment with high fidelity. The module considers several dimensions of fidelity, comparing task cue and response requirements for the fidelity dimensions with the cost of meeting these requirements to determine the dimensions for which increased fidelity is justified by increased training effectiveness. The output of this module is a set of possible training-device configurations applicable to the task set, each of which offers the greatest effectiveness for its cost.

Input requirements. The required input specifies fidelity dimensions and values, task cue and response requirements, and

output from the Instructional-Feature Selection Module, if available. Specific input requirements are enumerated below.

1. Fidelity dimension data. These data describe the range of options available for each fidelity dimension evaluated by the module. The variables include measures of technical performance and values used to estimate the development cost for each level of each fidelity dimension.

2. Task information processing requirements and activity data. These data are used by the fidelity rules to derive the task cue and response requirements. The task cue and response requirements describe the performance requirements for task cues and responses along each of the fidelity dimensions.

3. The results of the Instructional-Feature Selection Module, if they are available, may be incorporated in the module at the user's option.

4. Task training hours and costs. The time that would be required to train each task on actual equipment to the performance standard. For tasks in which training on actual equipment is impossible, the training time is an estimate of the training time required on a hypothetical training device with near-perfect fidelity and without any instructional features. The costs are normalized to produce task weights used in the analysis.

5. Fidelity Rules. These rules derive task cue and response requirements from task activity and information processing requirement data.

Module output. The output of this module is a range of training-device designs. Each design specifies a single level of sophistication on each fidelity dimension that is optimal in the sense that there are no other designs that are both lower in cost and greater in effectiveness. Once the set of optimal designs has been determined, a specific optimal design can be defined for user-specified cost or benefit levels.

Method. In the first step, the user selects the candidate fidelity dimensions and levels that will be evaluated in the model. In the second step, the cost and benefit of those fidelity levels are determined. The costs are calculated by the same estimating functions that are used in the Simulation Configuration Module, and represent development costs. The benefits are based on the number of tasks for which each level of the fidelity dimensions would be adequate, calculated by comparing the technical performance of each fidelity level with the cue and response requirements of the tasks for the appropriate fidelity dimensions. The resulting benefit measures are comparable across fidelity dimensions. In the third step, the optimal training device designs are determined based on the incremental benefit-to-cost ratios of the options. These designs may then be used by the Training-Device Selection or Resource Allocation Modules.

Training-Device Selection Module

This tool applies cost-effectiveness analysis to select the training devices that meet training requirements for all tasks at the minimum cost. The model considers (a) the fidelity of each training device compared to task cue and response requirements, (b) the instructional features present in each training device, and (c) the level of training conducted on each training device. The model recommends the optimal assignment of training devices to tasks, including (a) the devices that should be used to train each task, (b) the level of training at which each device should be employed, (c) the projected life-cycle cost of training, and (d) the level of performance expected to result from training for each selected device.

Input requirements. The major inputs to the model describe training requirements, device capabilities, and costs. The following types of data are required:

1. Task learning points. These data are numerical assessments of entry-level performance and the performance standard for each task.

2. Task activity data. These data are used by the fidelity rules to derive the task cue and response requirements. The task cue and response requirements describe the performance requirements for task cues and responses along each of the fidelity dimensions.

3. Task training hours and costs. These data estimate the time that would be required to train each task on actual equipment to the performance standard. For tasks in which training on actual equipment is impossible, the assessed training time would represent the training time required on a hypothetical training device with near-perfect fidelity and without any instructional features.

4. Task information processing characteristics. These data are used to estimate the effectiveness of the instructional features possessed by candidate training devices.

5. Training-device costs. These measures describe the development, fixed operating, and variable operating costs of the training device templates and actual equipment. Also included in these data are estimates of maximum annual utilization and the length of the device life cycle.

6. Training-device cue and response capabilities and instructional features. These data describe the fidelity and instructional features of both hypothetical training devices and device designs that were produced by the Fidelity Optimization Module.

7. Fidelity Rules. These rules derive task cue and response requirements from task activity data and information processing requirements.

8. Instructional feature rules. These rules associate characteristics of the tasks with instructional features.

9. Training-system data. These data include assumptions regarding the annual student throughput, the maximum effect of instructional features of efficiency, and the effect of simulation on setup time.

Module output. The outputs of the module include the optimal selection of training devices for each task, an estimate of the training criterion on each device for each task, and estimates of the overall life-cycle cost and training effectiveness of the optimal selection of training devices.

Method. In the first step of the analysis, cost and training effectiveness are calculated for each task to be trained, and the devices that meet the training requirements at the least cost are determined. Device effectiveness is measured by the expected increase in operational performance resulting from training on the device. Expected operational performance considers both the training criterion on the training device (expressed as a percentage of maximum performance) and the transfer of training from the training device to operational equipment. Transfer of training is determined primarily by comparing task cue and response requirements with device fidelity capabilities. Cost is determined by (a) the time required to learn the task, which, in turn, is influenced by the instructional features included on the devices, and (b) the hourly estimated costs.

In the second step, the individual task solutions are combined, and total costs and training time requirements are calculated. The analysis of individual tasks makes assumptions about the utilization of training devices that can only be verified after all tasks have been assigned to training devices. If the actual utilization is too different from the assumptions, the assumptions may be changed, and the analysis repeated at the user's option.

Resource Allocation Module

This module produces a detailed allocation of training resources to the training devices selected with the aid of the other tools. The objective of this module is to minimize the cost of meeting training requirements. The actual cost measure minimized is the pro rata per-student total training-system life-cycle cost incurred in training a student to criterion performance. The resources allocated refer to student training time at each stage of a course.

The Resource Allocation Module generalizes the Training-Device Selection Module in two ways: (a) It accounts for the

discontinuities in the cost function that arise from the procurement of individual training devices, (b) it allows the user to specify constraints on the total time a particular training device (or actual equipment) may be used, and on the minimum performance level at which a training device may be employed.

Input requirements. The input requirements for this tool include all those for the Training-Device Selection Module. In addition, the module requires the following constraints specified by the user.

1. User judgment regarding the maximum number of hours each type of training equipment would be available for use. This judgment reflects the total number of devices available. If the user does not specify any constraints, then the number of available devices will not be limited in the analysis.

2. User judgment regarding the minimum performance level at which each device may be used for each task. This constraint reflects the fact that for some tasks, practice on actual equipment (or possibly another device) might be too dangerous if the student did not already have sufficient skills.

Module output. The outputs of the module include the time each student should spend on each device, the number of devices that must be acquired to provide that time, and the final per-student pro rata costs of training to criterion.

Method. The method to solve the resource allocation problem is heuristic and iterative. It operates using detailed device cost curves made up of several linear segments. The general idea is to detail the device cost curves only around the solution. The strategy generates an initial solution using relatively undetailed cost curves. Then the cost curves are detailed around the initial solution and a new solution is found using the detailed cost curves. If the new solution lies within those cost curve domains, the process is terminated, and the solution is deemed optimal. If the current solution does not lie totally within current domains, the cost curves are re-detailed around the current solution, and a new cost solution found. If the process does not succeed after a predetermined set of iterations, the process is terminated and the last solution found is deemed "optimal," even though it lies outside the domain of detailing for one or more devices. In any case, the true cost functions are used to determine the per-student cost of training for the final solution.

The procedure for this module is carried out in three steps that are carried out iteratively. In the first step the training-device cost curves are represented as piecewise-linear functions. Because of limitations in processing time, the device cost curves are only partially detailed. In the second step the tasks are allocated to devices to minimize the total per-student training cost, as determined by the current set of cost curves. In the third step, the solution is examined to determine whether it is consistent with the cost curves. If the solution is consistent

with the cost curves, the process is complete. If an inconsistency exists, then the first step is carried out again, using the current solution as the basis for defining the cost curves.

Summary of Rule Bases

The numerical calculations performed by the OSBATS model are supported by two rule bases that employ reasoning processes that cannot easily be expressed numerically. The fidelity rule base and the instructional features rule base determine the values of intermediate parameters used by the OSBATS model. The fidelity rule base generates the task cue and response requirements from data about task activities. Similarly, the instructional features rule base generates a matrix that indicates which instructional features are appropriate for each task.

Structure of the Rule Base

The rule bases, developed using the EXSYS Expert System Development Package (copyright Exsys, Inc.), are composed of IF-THEN-ELSE logical statements called rules. The IF section of a rule contains the conditions which determine whether the rule will be executed and which section of the rule will be executed. The rule base evaluates the conditions in the IF section and determines whether they are true, false or unknown. If the conditions are true the THEN section is executed; if the conditions are false the ELSE section is executed. If the truth of the conditions is unknown the rule base searches for the necessary information before processing the rule.

The rule bases operate by a process called backward chaining, meaning that the rule base is searched for needed information before the user is asked to supply it. That is, the rule processor first examines rules that directly produce the desired conclusions. In the OSBATS rule bases, these rules assign tasks to instructional features or set cue and response requirements. If the conditions of the rules being examined are known to be true or false, then the appropriate inferences are made. If the truth of the conditions is unknown, then the rule processor searches for rules that can be used to derive the unknown values and attempts to apply these rules. This process continues until a rule is found that can be applied, or until all relevant rules have been examined. If required information is still unknown after all rules have been examined, the user is asked to supply the information.

Data in the rule bases are composed of qualifiers, variables, and choices. A qualifier is a qualitative variable that may take on one or more values from a specified set. It is represented as a multiple choice question. Qualifiers can be used in the IF section of a rule, or set the THEN and ELSE sections. Variables are most often used in mathematical and numerical expressions although they may have text values. Like qualifiers, variables can be used in the IF section, and set in the THEN and ELSE sections of a rule. Choices are the possible solutions and are assigned probability, or likelihood, values. For example, the Instructional

Features rule base contains as choices the twenty-one instructional features. In this rule base choices can be assigned likelihood values of zero or one; in other words, either the instructional feature enhances training or it does not.

Instructional Feature Rule Base

The instructional feature rules are a direct mapping of the task under consideration to appropriate instructional features. Data which the user must provide include the activities involved in the task, the task entry and standard performance levels, the time scaling factor (calculated from training times), the difficulty of the task (low, high), the probability of a crash (low, high) and several other factors involved in training a task. Two of the rules used in the Instructional Features rule base are shown below.

RULE NUMBER 1:

IF: The training device will be used to measure performance

THEN: Automated Performance Measurement - Probability = 1

RULE NUMBER 27:

IF: The type of activity involved in the task is continuous movement or procedures or decision making/rule using

and Discrete behaviors in this task are detectable by computer

THEN: The training device will be used to measure performance

ELSE: The training device will not be used to measure performance

Because Automated Performance Measurement has arbitrarily been made the first choice, rule number 1 is examined first. The rule base then searches the rules for information about whether the training device is used to measure performance. This information is found in rule number 27. To process rule 27 the rule base needs information about the type of activities involved in the task. Because the rule base contains no information on these activities, the user must supply the types of activities. If the first condition in rule 27 is true then the second condition is examined. The rule base now needs information about the discrete behaviors. Again, since this information is not available from the rule base the user must supply it. If both IF conditions are determined to be true the rule base will execute the THEN section of rule 27 and then process rule 1. Otherwise, the rule base will execute the ELSE section of rule 27 and skip rule 1.

Fidelity Requirement Rule Base

The fidelity requirement rule base evaluates task training and performance data to determine the cue and response requirements for each task across eleven fidelity dimensions. The rules in this rule base are grouped according to the fidelity dimension being evaluated. An explanation of each group of rules is given below, by dimension name. The first seven requirements, visual resolution, visual content, visual texture, front and side field-of-view, and points and area special effects, are calculated if an out-of-cockpit view is required. If no out-of-cockpit view is required these requirements are given the value zero.

Visual resolution. The visual resolution requirement involves the most lengthy calculations. The rule base calculates the visual resolution necessary to perform activities relevant to the task such as estimating altitude, slant range and clearance, detecting objects and landing on a slope. The visual resolution requirement for a task is determined by the activity that has the most severe resolution requirements.

Visual content. The value of the visual content requirement is determined by the level of minimal scene content that the user indicates is necessary to train the task effectively.

Visual texture. The visual texture requirement is calculated based on (a) the size of the topographical data base, (b) the environment in which the task is performed, and (c) whether judgments about altitude, slant range and clearance must be made to perform the task.

Front field-of-view. The front field-of-view requirement is calculated based on information about peripheral cues, operation of the aircraft when severely pitched up or rolled, whether identification of ground locations is a required activity and whether judgments of clearance must be made. This information determines the size of the front field-of-view. The size then determines the cue and response requirement.

Side field-of-view. The side field-of-view requirement is determined based on the required size of the side windows and whether both side windows are necessary. The size of the side windows is dependent on whether identification of ground locations is a required activity and whether the ground locations can be viewed through either the front or the side windows.

Special effects points requirement. The special effects requirement is based on the content elements that would enhance the effectiveness of training on the task. The content elements that the user can select from include moving ground vehicles, airborne vehicles, damaged vehicles, weapons blasts and cultural lights.

Special effects area requirement. The area effects requirement is also determined by the content elements that would

enhance training effectiveness. The possible area effects include rotor wash and smoke and dust.

Motion platform requirement. Platform motion is required instead of seat motion when (a) moderate or high cues for performance of the task are provided by one of the following two motions, lateral acceleration or yaw, and (b) either the motion cues are not correlated with visual cues or the motion provides a cue for the initiation of an emergency procedure. The value of the motion platform cue and response requirement is determined by the other motions needed to perform the task. The user can select from the following: longitudinal acceleration, lateral acceleration, vertical acceleration, yaw, pitch or roll.

Motion seat requirement. Seat motion is the default motion requirement when platform motion is not necessary. The seat motion cue and response requirement is calculated based on (a) which motions the user identifies as providing cues for performance of the task, (b) whether the type of activity involved in the task is NOT continuous movement, and (c) whether prolonged accelerations or decelerations are required.

Audio effects requirement. Determination of the audio effects requirement requires the user to identify the audio signals used to perform the task, indicate whether the signals are correlated with visual cues and indicate whether they provide a cue for the initiation of an emergency procedure.

Data base size requirement. The data base requirement is determined based on the minimum size of the topographical data base that the user indicates is necessary to perform the task. The required data base size is zero if there is no visual system, unless the task involves instrument flight.

Summary of Data Requirements

The OSBATS model requires a considerable amount of data to estimate optimal device designs and configurations. The specific data required and their format are derived from the methods and goals of the five modules. This section presents an overview of the input data requirements. The formal model description will specify and explain in detail how the data are used by the model. This summary briefly defines classes of data that are required, describes the required format, and discusses potential data sources.

There are two types of data required to support the functioning of the OSBATS model. The first type of data, called resident or internal data, cover the unchanging or slowly changing information and relational rules involved in the generation of options, tradeoffs, and configurations. The second type of data required by the model is situationally specific data, the data used to initiate execution of the models.

The resident data include the rules in the fidelity requirement and instructional feature rule bases, as well as descriptions of fidelity dimensions and levels, instructional features, cost estimates, assumptions used to determine learning parameters, and so forth. The input data are used to initiate execution of the models. These data include descriptions of the tasks to be taught, the task performance criteria to be met by the training, the number of students, and time required to train each task on actual equipment.

Although the model user is not considered the major provider of the data required by the model, certain inputs are required of the user. These inputs consist of the critical judgments that express general priorities in training-system design, and that limit the scope of the problem addressed by the OSBATS model. The user input requirements are enumerated below.

1. Weights that express the relative importance of operating-cost savings as opposed to safety and training-effectiveness concerns in determining whether a task should be trained on a training device or on actual equipment.

2. A value that reflects the relative importance of savings in investment cost as opposed to savings in operating cost.

3. Specification of whether all tasks should be weighted equally in the analysis, or whether tasks should be weighted according to the amount of training on actual equipment required to reach the training standard.

4. Specification of whether historical data on the likelihood that instructional features are used in existing training devices should be used in evaluating the benefit of these features in devices that are being designed.

5. Assumptions that should be made about training-device utilization to determine the total hourly cost of the training device.

6. Constraints on the maximum number of training devices to be procured, and on the minimum performance level in which each training device may be used for each task.

7. Limits on the tasks, training-device options, candidate instructional features, and fidelity options that are considered in the analysis.

Data Requirements and Format

The data requirements for the OSBATS modules can be organized into the following six categories with their respective subcategories.

1. Task training requirements. This class of data includes information about the training requirements associated with

the tasks that must be performed to prescribed standards following training. This class includes three subclasses.

a. Task learning points. These data describe for each task student entry performance level and performance standard on a scale that ranges from no knowledge (0) to expert performance levels (1.0).

b. Task simulation evaluation factors. These data include a rating of each task on a checklist of factors that are relevant to determining the need for simulation, including safety concerns, special performance conditions, and anticipated training effects.

2. Other task data. Other task data include three kinds of information about tasks.

a. Task training hours and costs. These data describe the training time and costs involved in meeting the training requirements for each task without a training device. Data elements describe the number of training hours required in classroom, actual equipment in both operational and non-operational modes, set-up time, and the cost of other required equipment.

b. Task information processing characteristics. These data rate tasks on a checklist of information-processing activities, such as timesharing or continuous-control processes, that are relevant to the evaluation of training-device instructional features.

c. Task activities. These data describe the activities required to perform the task according to the variables required by the fidelity rules. This class of data encompasses several variables that are specific to the task and domain.

3. Training-device data. This class of data describes hypothetical or actual training media in terms of cost, cue and response capabilities, and instructional features. This class includes three subclasses.

a. Training-device costs. These data include the following data elements for each training device: investment cost, annual fixed operating cost, hourly variable operating cost, maximum annual utilization, and training-device life cycle.

b. Training-device cue and response capabilities. These data rate the technical performance of each training device on each of the fidelity dimensions defined in 4.a.

c. Training-device instructional features. These data provide a checklist of the instructional features possessed by each training medium.

4. Fidelity dimension data. This class of data defines the set of options that are considered by the Fidelity Optimization Model, defines the technical performance scale in terms of concrete options, and contains parameters for estimating training-device cost as a function of cue and response capabilities. Fidelity dimension data include four subclasses.

   a. Fidelity dimensions and levels. These data define each fidelity dimension and list the names of all levels and the associated technical performance rating, on a scale from 0 to 1.0.

   b. Fidelity dimension cost data. This class of data includes the three parameters of the function that is used to estimate the cost of a particular level from its technical performance. The three parameters describe the minimum cost, maximum cost, and an exponent that describes the shape of the cost curve.

   c. Fidelity dimension minimum performance parameter. This parameter, assessed for each fidelity dimension, estimates the transfer of training that would occur when the capability on the subject fidelity dimension is nil, but capabilities on all other dimensions are perfect.

   d. Fidelity rules. These data are an ordered set of conditional statements that derive the task cue and response requirements from a description of the activities required to perform a task.

5. Instructional-feature data. This class of data describes the costs and benefits of the instructional features and gives specific rules for associating instructional features to tasks. Included in this class of data are two subclasses.

   a. Instructional-feature rules. Instructional feature rules specify the conditions under which each instructional feature would improve training efficiency. The conditions may reference other elements in the data base.
   b. Instructional-feature cost and weight. The data elements in this class include an assessment of the development cost of each instructional feature, and an assessed weight that moderates the calculated benefit values for instructional features.

6. Training-system data. This class of data includes a variety of miscellaneous data and general information about the training course. This data class includes the following two subclasses.

   a. Course and system information. A single element describing the required number of graduates per year is included in this category.

b. Model information. This class of data includes a variety of assumptions used by the model. The nature of each data element is described in the formal model description.

Each of the categories described above consists of several data elements. Table 26 gives a brief description of each data element, and lists the scale on which it is measured. Data elements will be referenced by the category number listed in the table. For example, the set of performance standards will be referred to as "1A1." Similarly, a class of data elements will receive the appropriate number of the general data category for that class. For example, task training requirements will be referred to by the category "1." Also included in Table 26 are potential sources of the data; the nature of these sources is described in the following paragraphs.

<u>Data Sources</u>

The data required to operate the OSBATS model will come from several sources, including:

1. subject-matter experts (SMEs),
2. training-system experts (TSEs),
3. training researchers (TRs),
4. model developers (MDs), and
5. model users (MUs).

As the model evolves, we expect the nature of the data required from experts to change, with subject-matter and training-system experts providing simpler judgments that are more factual and less subjective. These judgments would be transformed to produce the data required by the model. In the near term, however, experts will be required to provide a variety of judgmental data to meet the model requirements.

The likely source of each data element is shown in Table 1. General descriptions of these data sources are given below.

1. Subject matter experts include instructors, training developers, and expert job performers. These experts are characterized by their knowledge of the tasks being trained. They are the primary source of task training requirement and other task data.

2. Training-system experts are characterized by their knowledge of the capabilities and costs of training devices. They are the primary source of training-device data, fidelity dimension data, and instructional-feature cost data.

3. Training researchers provide the link between the model and the body of relevant behavioral research. They will be the major source of instructional-feature data. In addition, training research will play an important part in the form of the functions used to predict training cost and effectiveness.

Table 1

Description of OSBATS Model Data Requirements

DB Cat.	Description	Format/ Scale	Source
1	Task Training Requirements		
1A	Task Learning Points		
1A1	Entry Performance Level	[0,1.0]	SME
1A2	Performance Standard	[0,1.0]	SME
1B	Task Simulation Evaluation Factors		
1B1	Absolute Requirement	{0,1}	SME,MU
1B2	Special Weather	{0,1}	SME,MU
1B3	Special Situation	{0,1}	SME,MU
1B4	Special Equipment	{0,1}	SME,MU
1B5	Training Effectiveness Enhancements	{0,1}	SME,MU
2	Other Task Data		
2A	Task Training Hours and Costs		
2A1	Classroom Hours	hours/grad	SME
2A2	Actual Equip Non-Operational Hours	hours/grad	SME
2A3	Actual Equipment Operational Hours	hours/grad	SME
2A4	Actual Equipment Setup Hours	hours/grad	SME
2A5	Other Training Costs	$/grad	SME
2B	Task Info Processing Characteristics		
2B1	Continuous Movement	{0,1}	SME,TR
2B2	Procedural	{0,1}	SME,TR
2B3	Perception	{0,1}	SME,TR
2B4	Decision Making/Rule Using	{0,1}	SME,TR
2B5	Verbal Information	{0,1}	SME,TR
2B6	Voice Communication	{0,1}	SME,TR
2B7	More Than Seven Steps	{0,1}	SME,TR
2B8	Time Sharing	{0,1}	SME,TR
2B9	Computer-Detectable Responses	{0,1}	SME,TR
2B10	Meaningful Performance Tolerances	{0,1}	SME,TR
2B11	Intrinsic Feedback	{0,1}	SME,TR
2B12	Cue Salience	{0,1}	SME,TR
2B13	Crash Probability	{0,1}	SME,TR
2B14	Situational Awareness	{0,1}	SME,TR
2C	Task Activities	various	SME
3	Training Device Data		
3A	Training Device Costs		
3A1	Investment Cost	$/device	TSE
3A2	Annual Fixed Operating Cost	$/device/yr	TSE
3A3	Hourly Variable Operating Cost	$/grad/hr	TSE
3A4	Lifecycle	years	TSE,MU
3A5	Maximum Annual Utilization	hrs/yr	TSE,SME

Table 1 (Continued)

Description of OSBATS Model Data Requirements

DB Cat.	Description	Format/ Scale	Source
3B	Training Device Cue & Response Capabilities		
3B1	Visual Resolution	[0,1.0]	TSE
3B2	Visual Content	[0,1.0]	TSE
3B3	Visual Texture	[0,1.0]	TSE
3B4	Front Visual Field of View (FOV)	[0,1.0]	TSE
3B5	Side Visual FOV	[0,1.0]	TSE
3B6	Point Special Effects	[0,1.0]	TSE
3B7	Area Special Effects	[0,1.0]	TSE
3B8	Platform Motion	[0,1.0]	TSE
3B9	Seat Motion	[0,1.0]	TSE
3B10	Sound Special Effects	[0,1.0]	TSE
3B11	Map Area	[0,1.0]	TSE
3C	Training-Device Instructional Features		
3C1	Automated Performance Measurement	{0,1}	TSE
3C2	Performance Indicators	{0,1}	TSE
3C3	Procedure Monitoring	{0,1}	TSE
3C4	Automated Performance Alerts	{0,1}	TSE
3C5	Augmented Feedback	{0,1}	TSE
3C6	Augmented Cues	{0,1}	TSE
3C7	Record/Playback	{0,1}	TSE
3C8	Total System Freeze	{0,1}	TSE
3C9	Remote Graphics Replay	{0,1}	TSE
3C10	Initial Conditions	{0,1}	TSE
3C11	Scenario Control	{0,1}	TSE
3C12	Crash Override	{0,1}	TSE
3C13	Reset/Reposition	{0,1}	TSE
3C14	Parameter Freeze	{0,1}	TSE
3C15	Flight System Freeze	{0,1}	TSE
3C16	Positional Freeze	{0,1}	TSE
3C17	Real-Time Variables Control	{0,1}	TSE
3C18	Automated Simulator Demonstration	{0,1}	TSE
3C19	Adjunct CAI	{0,1}	TSE
3C20	Automated Adaptive Training	{0,1}	TSE
3C21	Automated Cueing and Coaching	{0,1}	TSE
4	Fidelity Dimension Data		
4A	Fidelity Dimensions and Levels		
4A1	Technical Performance Levels	[0,1.0]	MD,TSE,SME
4B	Fidelity Dimension Cost Data		
4B1	Minimum Cost	investment $	TSE,MD
4B2	Maximum Cost	investment $	TSE,MD
4B3	Exponent	>0	TSE,MD
4C	Minimum Performance Parameter	[0,1.0]	TR,TSE,SME
4D	Fidelity Rules	not numeric	TR,MD,SME

Table 1 (Continued)

Description of OSBATS Model Data Requirements

DB Cat.	Description	Format/ Scale	Source
5	Instructional Feature Data		
5A	Instructional Feature Rules	not numeric	TR,MD
5B	Instructional Feature Cost and Weight		
5B1	Instructional Feature Cost	investment $	TSE
5B2	Instructional Feature Benefit Weight	[0,1.0]	TR,SME
6	Training System Data		
6A	Course and System Information		
6A1	Annual Student Throughput	grads/yr	SME,MU
6B	Model Information		
6B1	Standard Adjustment Rate	%	TR,MD
6B2	Learning Curve Exponent	>0	TR,MD
6B3	Maximum Instructional Feature Effect	%	TR,MD
6B4	Maximum Number of Instructional Features	{1,2,3,...}	TR,MD
6B5	Assumed Setup Savings Percentage	%	TR,MD
6B6	Cost Savings Weight	[0,1.0]	MU
6B7	Recommendation Boundaries	[0,1.0]	MD
6B8	Annual Fixed Cost Factor	$inv/$fixed	TSE,MD
6B9	Hourly Variable Cost Factor	$inv/$var	TSE,MD
6B10	New Device Utilization	hours/year	TSE,MD
6B11	New Device Lifecycle	years	TSE,MD

4. Model developers are required to produce data in some of the areas in which consideration of the model form is required. For example, the fidelity optimization module assumes that cost and benefit of fidelity dimensions are mutually independent. The model developer will be able to structure the fidelity dimensions to reduce or eliminate the effect of any interactions. Consequently, we expect that the model developer will work with the training-system expert to define the fidelity dimensions and levels.

5. The model user may be one of the other four kinds of experts, or may be a project manager who must aggregate the specific expertise of members of his staff. Although the user will have access to the data base, we expect the user to have three major impacts: (a) to make the value judgments that affect the critical weights used at various points in the analysis, (b) to set the scope of the analysis, and (c) to adjust the results of the analysis to account for factors that are not included in the model.

Summary of Input Requirements for Individual Modules

Table 2 summarizes the input requirements for the five OSBATS modules. For each data element, it indicates the modules for which the information is required.

Summary of Model Assumptions

The OSBATS model makes several assumptions regarding the training process. Some of these assumptions are central to the model's description of the training process. Other assumptions simplify the problem so that it can be addressed efficiently with minimal input data. Still other assumptions represent conjectures about aspects of the training process for which there is little empirical data. Details of these assumptions are given in the formal system description. This section describes some of the more important model assumptions.

The model assumptions are organized into the following four classes.

1. Assumptions that specify the functional relationship between variables, such the relationship between performance and amount of training.

2. Assumptions that specify that certain variables are independent of other variables.

3. Assumptions that deal with the model's treatment of cost.

4. Specific assumptions about parameter values and measurement scales.

These classes of assumption are described in the following subsections.

Table 2

Module Use of Data Elements

DB Cat.	Description	SC	IFS	FO	TDS	RA
		Optimization Module: Simulation Configuration	Instr. Feature-Selection	Fidelity Optimization	Training-Device Selection	Resource Allocation
1	Task Training Requirements	x	x	x	x	x
1A	Task Learning Points	x	x		x	x
1A1	Entry Performance Level	x	x		x	x
1A2	Performance Standard	x	x		x	x
1B	Task Simulation Evaluation Factors	x				
1B1	Absolute Requirement	x				
1B2	Special Weather	x				
1B3	Special Situation	x				
1B4	Special Equipment	x				
1B5	Training Effectiveness Enhancements	x				
2	Other Task Data	x	x	x	x	x
2A	Task Training Hours and Costs	x	x	x	x	x
2A1	Classroom Hours	x	x	x	x	x
2A2	Actual Equip Non-Operational Hours	x	x	x	x	x
2A3	Actual Equipment Operational Hours	x	x	x	x	x
2A4	Actual Equipment Setup Hours	x	x	x	x	x
2A5	Other Training Costs	x	x	x	x	x
2B	Task Info Processing Characteristics	x	x		x	x
2B1	Continuous Movement	x	x		x	x
2B2	Procedural	x	x		x	x
2B3	Perception	x	x		x	x
2B4	Decision Making/Rule Using	x	x		x	x
2B5	Verbal Information	x	x		x	x
2B6	Voice Communication	x	x		x	x
2B7	More Than Seven Steps	x	x		x	x
2B8	Time Sharing	x	x		x	x
2B9	Computer-Detectable Responses	x	x		x	x
2B10	Meaningful Performance Tolerances	x	x		x	x
2B11	Intrinsic Feedback	x	x		x	x
2B12	Cue Salience	x	x		x	x
2B13	Crash Probability	x	x		x	x
2B14	Situational Awareness	x	x		x	x
2C	Task Activities	x		x	x	x

Table 2 (Continued)

Module Use of Data Elements

Optimization Module

DB Cat.	Description	SC (Simulation Configuration)	IFS (Instr. Feature-Selection)	FO (Fidelity Optimization)	TDS (Training-Device Selection)	RA (Resource Allocation)
3	Training Device Data	x	x	x	x	x
3A	Training Device Costs	x	x	x	x	x
3A1	Investment Cost	x	x	x	x	x
3A2	Annual Fixed Operating Cost	x	x	x	x	x
3A3	Hourly Variable Operating Cost	x	x	x	x	x
3A4	Lifecycle	x	x	x	x	x
3A5	Maximum Annual Utilization	x	x	x	x	x
3B	Training Device Cue & Response Capabilities	x			x	x
3B1	Visual Resolution	x			x	x
3B2	Visual Content	x			x	x
3B3	Visual Texture	x			x	x
3B4	Front Visual Field of View (FOV)	x			x	x
3B5	Side Visual FOV	x			x	x
3B6	Point Special Effects	x			x	x
3B7	Area Special Effects	x			x	x
3B8	Platform Motion	x			x	x
3B9	Seat Motion	x			x	x
3B10	Sound Special Effects	x			x	x
3B11	Map Area	x			x	x
3C	Training-Device Instructional Features	x			x	x
3C1	Automated Performance Measurement	x			x	x
3C2	Performance Indicators	x			x	x
3C3	Procedure Monitoring	x			x	x
3C4	Automated Performance Alerts	x			x	x
3C5	Augmented Feedback	x			x	x
3C6	Augmented Cues	x			x	x
3C7	Record/Playback	x			x	x
3C8	Total System Freeze	x			x	x
3C9	Remote Graphics Replay	x			x	x
3C10	Initial Conditions	x			x	x
3C11	Scenario Control	x			x	x
3C12	Crash Override	x			x	x
3C13	Reset/Reposition	x			x	x
3C14	Parameter Freeze	x			x	x
3C15	Flight System Freeze	x			x	x
3C16	Positional Freeze	x			x	x
3C17	Real-Time Variables Control	x			x	x
3C18	Automated Simulator Demonstration	x			x	x

Table 2 (Continued)

Module Use of Data Elements

DB Cat.	Description	SC	IFS	FO	TDS	RA
		Optimization Module: Simulation Configuration	Instr. Feature-Selection	Fidelity Optimization	Training-Device Selection	Resource Allocation
3C19	Adjunct CAI	x			x	x
3C20	Automated Adaptive Training	x			x	x
3C21	Automated Cueing and Coaching	x			x	x
4	Fidelity Dimension Data	x		x	x	x
4A	Fidelity Dimensions and Levels	x		x		
4A1	Technical Performance Levels	x		x		
4B	Fidelity Dimension Cost Data	x		x		
4B1	Minimum Cost	x		x		
4B2	Maximum Cost	x		x		
4B3	Exponent	x		x		
4C	Minimum Performance Parameter	x			x	x
4D	Fidelity Rules	x		x	x	x
5	Instructional Feature Data	x	x		x	x
5A	Instructional Feature Rules	x	x		x	x
5B	Instructional Feature Cost and Weight		x			
5B1	Instructional Feature Cost		x			
5B2	Instructional Feature Benefit Weight		x			
6	Training System Data	x			x	x
6A	Course and System Information	x			x	x
6A1	Annual Student Throughput	x			x	x
6B	Model Information	x			x	x
6B1	Standard Adjustment Rate	x			x	x
6B2	Learning Curve Exponent	x			x	x
6B3	Maximum Instructional Feature Effect	x			x	x
6B4	Maximum Number of Instructional Features					
6B5	Assumed Setup Savings Percentage	x			x	x
6B6	Cost Savings Weight	x				
6B7	Recommendation Boundaries	x				
6B8	Annual Fixed Cost Factor			x		
6B9	Hourly Variable Cost Factor			x		
6B10	New Device Utilization			x		
6B11	New Device Lifecycle			x		

Functional Relationships

The OSBATS model includes several functions that relate cost and benefit measures to training time, task characteristics, and training-device features. This section gives a verbal description of these assumed relationships. A mathematical statement of the assumptions is given in the formal model description and in Appendix B.

Central to the Training-Device Selection Module and the Resource Allocation Module is a relationship between performance on actual equipment and training time (on a training-device or on actual equipment). This relationship is also used in the Simulation Configuration Module. The OSBATS model assumes that this relationship is characterized by a power function. Newell and Rosenbloom (1981) have reviewed a large body of literature indicating that a power function provides a good fit for data for which the time to complete a task is related to the number of practice trials. A mathematical analysis of the power function indicates that it is equally appropriate when training is measured by time instead of trials. We have extended the power function to apply to a more general measure of performance. The resulting function incorporates learning on the device in which training is conducted, and subsequent transfer of the training to actual equipment.

A critical parameter of the performance function is its asymptote, which represents the maximum extent to which training on a training device will transfer to actual equipment. Transfer of training depends on the training device being used to provide the cue and response requirements of the tasks being trained. Thus, it a function of training device fidelity. The OSBATS model treats this relationship as a product of training-device capability measures over the fidelity dimensions. The factors of the product function are assessed so that different fidelity dimensions receive different importance weights in the relationship, and so that fidelity dimensions for which a task requirement is high are more important in the assessment for that task than dimensions in which the requirement is low. The asymptote function includes normalization constants that ensure that the results of the function are in the correct range, and are consistent with the assessed cue and response requirements.

The Fidelity Optimization Module and Simulation Configuration Module rely on a function that estimates training-device development cost from training-device fidelity and instructional features. A critical component of this function estimates the cost of reaching a particular technical performance level on a given fidelity dimension. The OSBATS model characterizes the development-cost component for a single fidelity dimension as a logarithmic function of technical performance. This function has been formulated to have the reasonable property that increments in technical performance at a high level of fidelity are more expensive than increases at a low level of fidelity. Each fidelity

dimension is described by a single number characterizing the specific shape of the cost function.

The Training-Device Selection Module and Resource Allocation Module must have a measure of the hourly life cycle cost of training devices. The Training-Device Selection Module considers a single hourly rate for each training device, combining investment, fixed operating, and variable operating cost components and dividing by the estimated total use of the training device over its life cycle. A single cost rate leads to a cumulative cost that is a linear function of the number of hours that the training device is used. The Resource Allocation Modules uses a piecewise linear cumulative cost function with discontinuities at the levels of use that require the procurement of additional training devices.

Independence Assumptions

Independence assumptions simplify the OSBATS model by ignoring some of the many interactions that are possible between training system variables. To the extent that the assumptions are reasonable, independence can simplify the analysis process and add to the clarity of the results. However, if the assumptions are not correct, then the method will not identify the best solutions. We have assumed independence in several areas, as enumerated below:

1. The cost and benefit of any specific instructional feature does not depend on which other instructional features are present in a training device.

2. The cost and benefit of a particular level of a fidelity dimension does not depend on the levels of other dimensions in a training device.

3. The fidelity of a training device is assumed to affect the asymptote of the performance function (transfer of training) without affecting the rate at which performance approaches that asymptote.

4. The instructional features of a training device are assumed to affect the rate at which the performance approaches the asymptote of the performance function, but does not affect the asymptote itself.

5. The effectiveness of a training device for a particular task depends only on the performance level attained before the device was used, and does not depend on which device or devices were used to attain that performance level.

6. Effects of the sequence in which tasks are trained are not considered in the OSBATS model. The assessed time required for training assumes that tasks are trained in a reasonable order.

Cost Component Assumptions

Assumptions of specific cost estimating functions were discussed previously. The major concern of this section is how the different modules treat the one-time and recurring components of life cycle cost. In this respect, there are differences between the modules that stem from differences in the aspect of the overall problem that they are addressing. The Training-Device Selection and Resource Allocation Modules are directly concerned with minimizing the life cycle cost of training. Consequently, these modules use life cycle cost as their measure of cost. The Instructional Feature Selection and Fidelity Optimization Modules are concerned with the selection of training-device components. The development cost of each component may be specified in advance. However, the operating cost depends on the number of tasks for which the component would be required. Thus for these modules, operating cost is confounded with benefit. Consequently, the Instructional Feature Selection and Fidelity Optimization Modules consider development cost as the major cost measure. The Simulation Configuration Module considers both development cost and operating cost in its analysis. Development cost is the major cost element, while operating cost savings is explicitly included as one of the benefits of simulated training.

Specific Assumptions

The OSBATS model requires estimates for several parameters for its operation. Also included among these assumptions are scaling assumptions on the measurement of benefit. The specific assumptions are the following:

1. In the Simulation Configuration, Instructional Feature Selection, and Fidelity Optimization Modules, benefit is measured on a ratio scale. This assumption allows the model to set the scale of benefit so that it varies from 0.0 to 1.0 among the options being considered in the particular analysis.

2. The model includes parameters stating the maximum number of instructional features that would be useful in training a single task and the maximum percentage improvement in training time that could be obtained from the inclusion of instructional features. These values are currently assumed to be three features and 10%, respectively.

3. The model includes a parameter expressing the percentage of setup time that can be saved by training in a simulated environment. That number is currently assumed to be 60%.

Whereas many of the functional relationships described previously represent assumptions at the heart of the OSBATS model, these specific assumptions represent specific parameter values that could be changed with minimal effects on model operation.

OSBATS MODEL DESCRIPTION

The detailed model description is presented using IDEF0 (Integrated Computer Aided Manufacturing Definition). The IDEF0 system was developed by the Integrated Computer Aided Manufacturing Office (ICAM) of the U.S. Air Force to be used as a tool for describing the functions and data of a complex system (SofTech, Inc., 1981; Ross & Schoman, 1977). In this report, IDEF0 is used to describe the components and functions of the implemented OSBATS model. The first part of this section provides a conceptual description of the IDEF0 system. The next portion provides tables that summarize information about the model nodes, input and control data, and process outputs. Also included is a list of the symbols used to represent the variables of the model. Finally, the IDEF0 description of the model components are presented in an organized node structure.

IDEF0 Methodology Description

A system consists of any combination of machinery (hardware), data, and people, working together to perform a useful function. IDEF0 is a technique that enables people to understand complex systems and to communicate their understanding to others. IDEF0 describes the functions performed by the system by successively decomposing the system into its basic components, describing how each component processes information, and specifying how different components interact. An IDEF0 model is expressed as a series of related diagrams; each diagram describes a particular system component or function. An IDEF0 diagram is composed of boxes and arrows. The boxes represent component functions or activities, while the arrows represent data that affect the activities or are produced by them.

IDEF0 Model Organization

The diagrams in an IDEF0 model describe the system in a modular, top-down fashion, showing the breakdown of the system into its component parts. The application of IDEF0 starts with the most general or abstract description of the system to be produced. This description is represented in a diagram as a single box; that box is subsequently broken down into a number of more detailed boxes, each of which represents a component part. The component parts are then detailed, each on another diagram. Each part shown on a detail diagram is again broken down, and so forth, until the system is described to the desired level of detail. Lower-level diagrams, then, are detailed breakdowns of higher-level diagrams. At each stage of breaking down the system, the higher-level diagram is said to be the "parent" or overview of the lower-level "detail" diagrams. The relationship between diagrams at different levels is shown in Figure 1.

Diagram display format. In this document, each diagram in an IDEF0 model is displayed in a two-page format. The subject diagram is shown on the top of the right-hand page. The parent of the subject diagram is shown on the top of the left-hand page

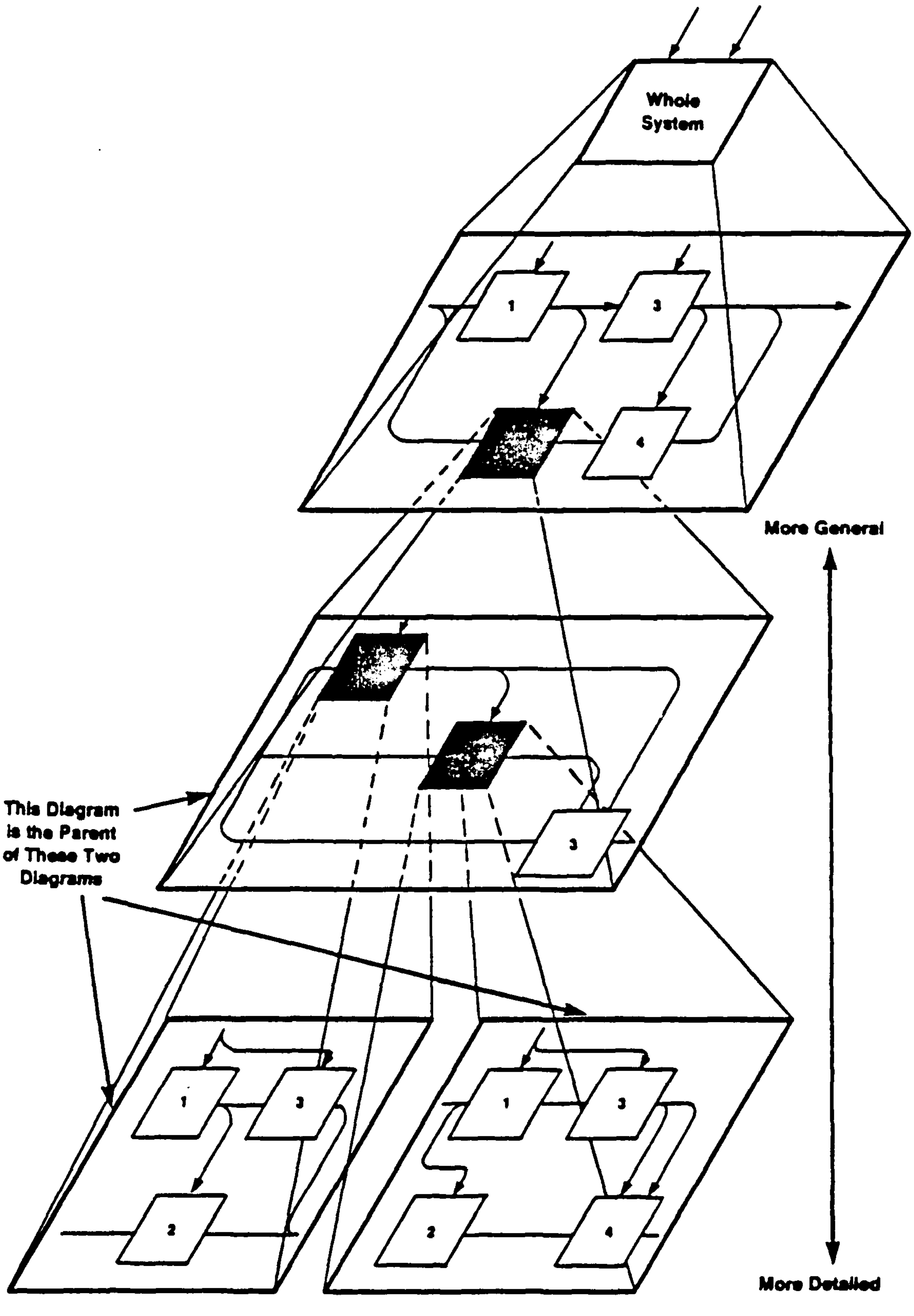


Figure 1. Example of a hierarchical, top-down model.

with the location of the subject node indicated. On the bottom half of both pages is text describing the operations performed by each activity represented in the diagram. Each pair of pages receives a page number that is displayed as part of the subject diagram.

Diagram node numbers. In an IDEF0 diagram, the component parts are shown as numbered boxes. A diagram should have no more than six boxes. Each box at one level is detailed in one diagram at the next lower level until a sufficient level of detail is reached. The place of each diagram in a model is indicated by a "node number" derived from the numbering of boxes. For example, A21 is the diagram that details box 1 on the A2 diagram. Similarly, A2 details box 2 on the A0 diagram, which is the top diagram of the model. The parent of the A0 diagram represents the system as a single box and is denoted "A-0." The hierarchy may be shown in an index of diagram names and their node numbers called a "node list." The node list serves as a table of contents for a model. In an IDEF0 model, diagrams are displayed according to the order of their node numbers.

The example shown in Figure 6 provides an illustration of the hierarchical decomposition of functions. The diagrams in Figure 6 indicate that the overall function, develop system (A0), is broken down into three sub-functions, A1 through A3. Design system (A2) is further broken down into three, more detailed sub-functions (A21 through A23).

Description of Individual IDEFO Diagrams

In IDEF0, boxes represent components in the breakdown, and arrows represent relationships between these components. Descriptive labels are written inside each box and along each arrow to describe their meaning. The notation is kept simple to permit easy reading with little special training.

Figure 7 shows a sample IDEF0 diagram. Notice that the boxes represent the breakdown of activities or functions performed by the system and are named by verbs. Arrows, which represent objects or information, are labeled with nouns.

Box-and-arrow syntax. The sample IDEF0 diagram in Figure 7 shows that the descriptive names and labels convey the box and arrow contents to the reader. In addition to its label, the side at which an arrow enters or leaves a box shows its role as an input, control, output, or mechanism for the box (see Figure 8). Arrows that enter from the left of an activity box represent inputs to the process represented by the box. Inputs represent the raw materials or data used by the activity to produce outputs. The outputs are represented by arrows that originate from the right side of the box. Arrows entering a box from the top represent controls on the activity. Controls are data that provide catalysts or constraints for the represented activity, but are not changed by the process. Finally, arrows that enter a

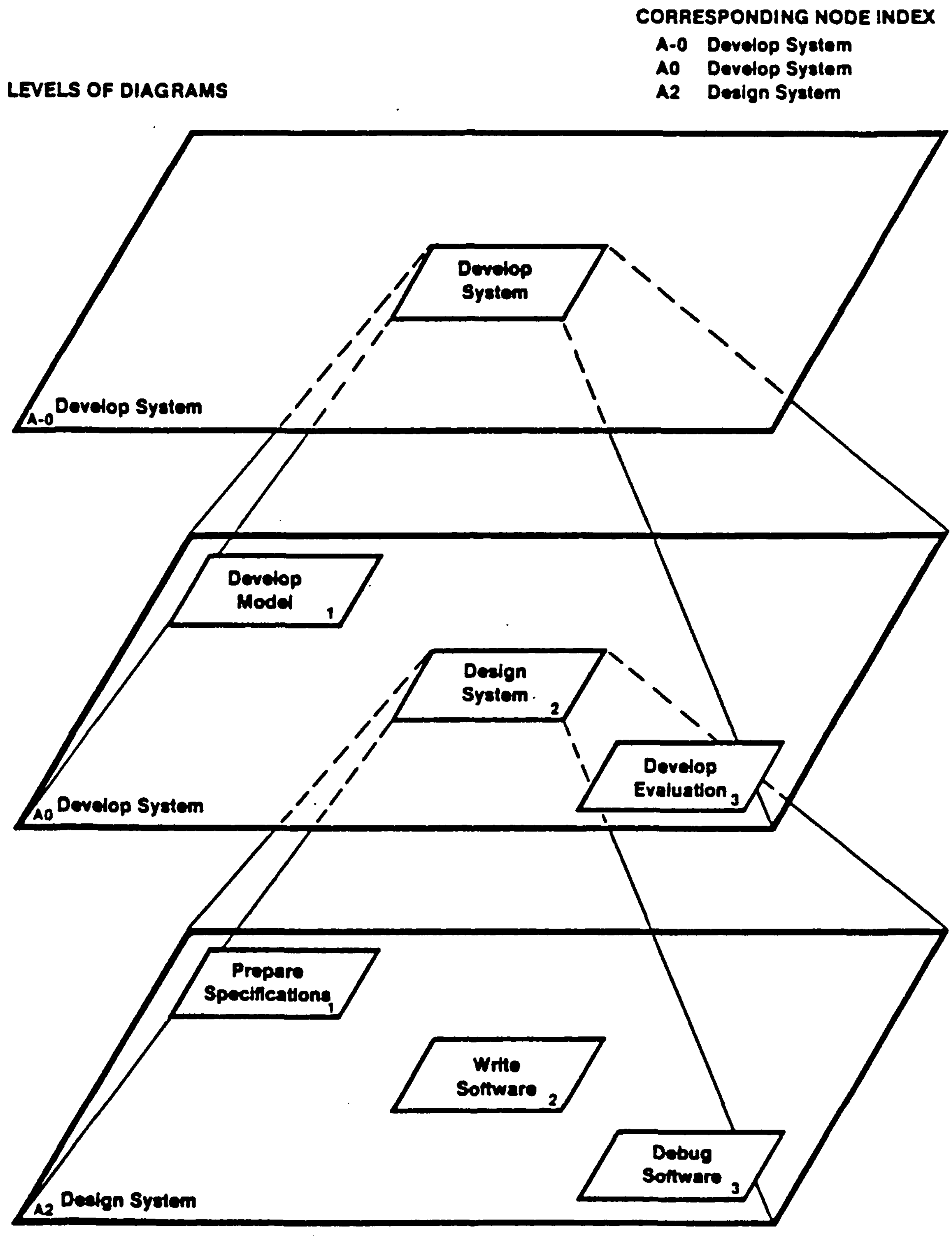


Figure 2. IDEF0 node numbering convention.

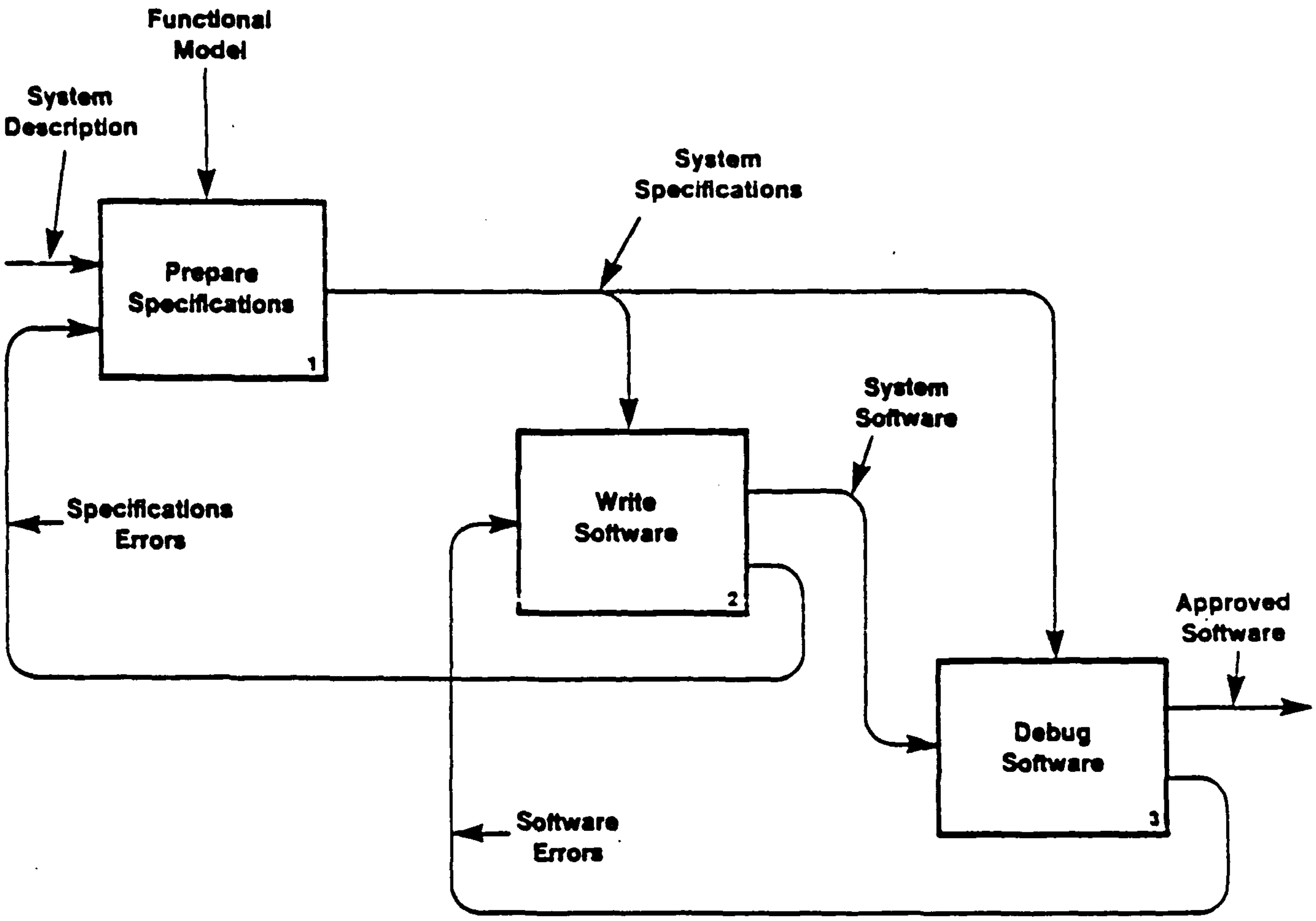


Figure 3. Sample IDEF0 diagram.

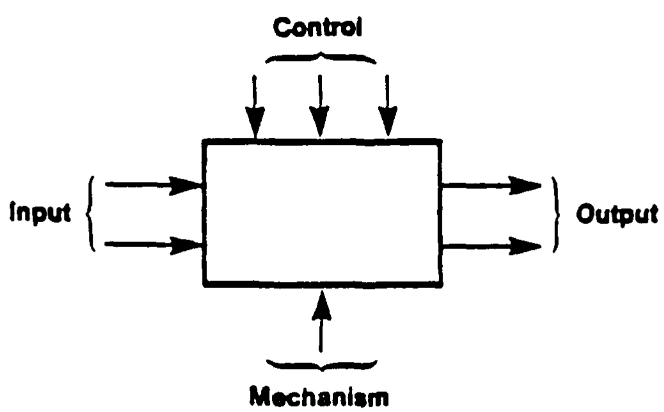


Figure 4. Sample IDEF0 diagram showing box and arrow syntax.

box from the bottom represent mechanisms. Mechanisms are the agents that perform the activities represented in the box. In short, inputs and outputs represent what is done by the process, controls represent why it is done, and mechanisms represent how it is done.

The arrow structure of an IDEF0 diagram represents a constraint relationship among boxes. It does not represent flow of control or sequence. The arrows entering a box show all that is needed by the box to perform its function. Therefore, the box is constrained by its inputs and controls.

Labeling of arrows. Some arrows show both their source and destination boxes on the same diagram, while others have one end unconnected (see Figure 5). The unconnected arrows represent inputs, controls, or outputs of the parent box. To find the source or destination of these unconnected arrows, the reader must locate the matching arrows on the parent diagram. All such unconnected arrows must continue on the parent for the diagrams to be complete.

Although arrow connections from parent boxes to detail diagrams is sometimes obvious from the labels, we have developed a special notation that should allow readers to do the match quickly. The notation used to describe the OSBATS model is slightly different from standard IDEF0 procedures for labeling unconnected arrows. The data for the OSBATS model are described in a structured data base. Each element in the data base is identified by a unique outline number (e.g., 2A1). Input and control arrows that represent data in the data base are labeled with the appropriate outline number. Often it is the case that data are described more generally at higher-level nodes than they are at lower-level nodes. Thus, a particular input or control may be labeled "2" at node A0, "2A" at node A1, and "2A1" at node A13.

A somewhat different labeling scheme is used for output arrows. Output arrows are labeled according to the highest-level node at which the output originated. For example, the output of node A212 will be labeled O212A if it does not occur at any higher-level node. If there are three outputs for A212, they will be labeled O212A, O212B, and O212C. The label is consistent across all nodes in which the output is represented. Therefore, if the first output for node A212 is also shown at node A21234, it will still be labeled O212A. Occasionally, the same output is represented at higher- and lower-level nodes, but it is more detailed at the lower-level node. When this occurs, the output will retain the node number of the higher-level node but will receive an additional number to represent the division of the output into parts. For example, the output O212A may be represented as O212A1, O212A2, and O212A3 at a lower-level node. If one of these outputs is further subdivided at a lower-level node, it will receive a second letter. For example, if O212A2 is divided into three components, the components will be referenced as O212A2A, O212A2B, and O212A2C.

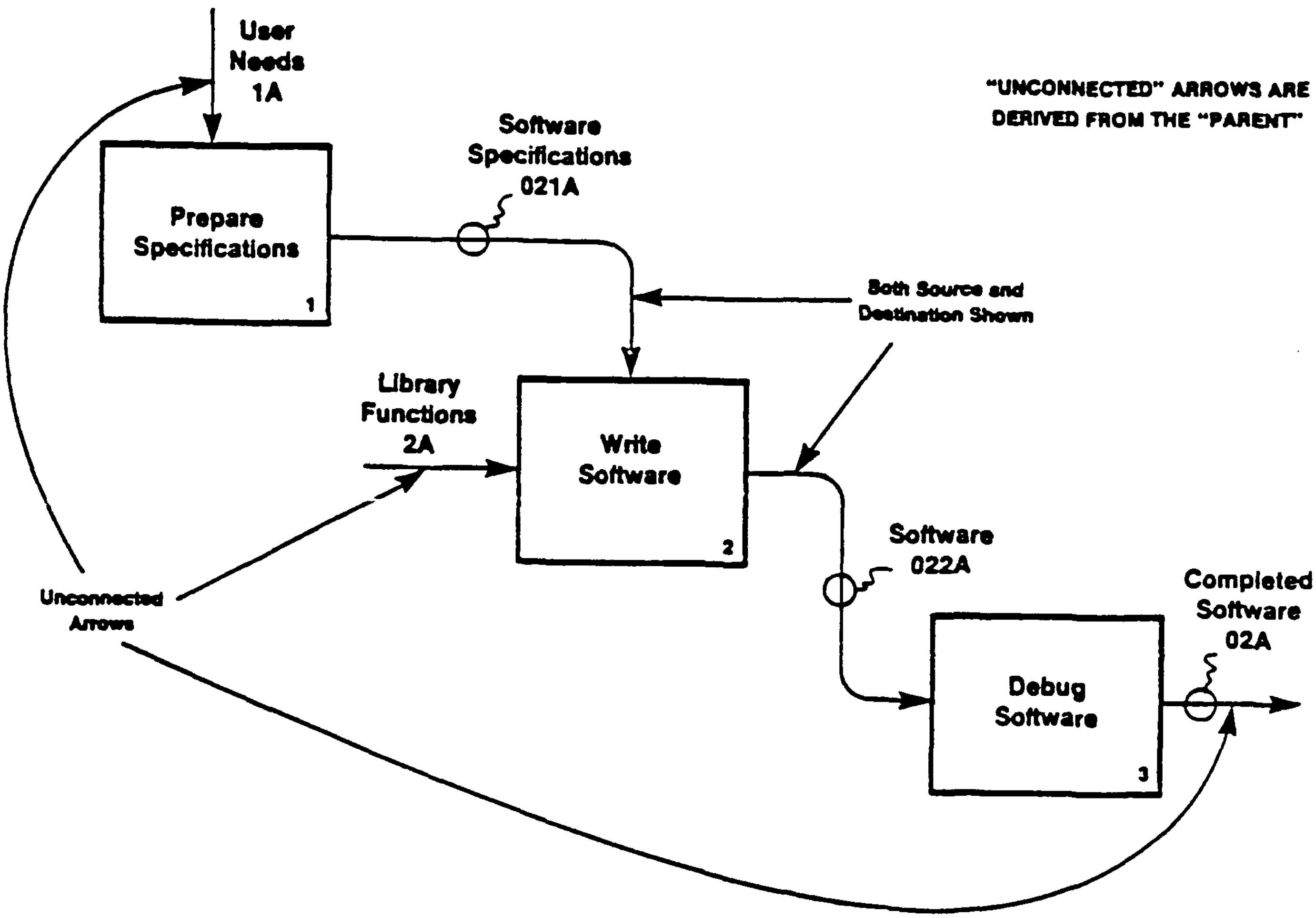


Figure 5. Sample IDEF0 diagram showing source and destination.

Mechanism arrows are used sparingly in the OSBATS model definition. When they are used, their reference is clear. Consequently, the mechanism arrows are not numbered and are identified only by their label.

It is possible for a data element to serve as an input to some sub-activities of a given activity and as a control for other sub-activities. In this case, the data will be represented once in the parent diagram, either as input or control. In the detailed diagrams, the data would be represented as a control in some diagrams and as an input in others, as appropriate.

Nodes, Data, and Variables

This section presents three tables that summarize information about the model nodes, input and control data, and process outputs. Also included is a list of the symbols used to represent the variables of the model. Specifically, the following tables are included.

Table 3 provides a list of the nodes in the structure in the order that they appear in the system description. The node list provides the table of contents for the IDEF0 model. If the node is represented by its own detailed diagram, the number of that diagram is listed in the final column of the node list. Nodes that have no detailed diagram do not have an IDEF diagram number listed. The descriptions of such a node may be found on the diagram for its parent node.

Table 4 provides a data base map that enumerates all data requirements. The data base map lists all node numbers in which each data element data is used, and specifies whether the data are used as an input or a control to each node.

Table 5 provides a list of the model outputs indicating the source node and all destination nodes for each output. A destination listed as "UP" indicates that the output is unconnected in a particular IDEF diagram. The output map specifies whether each output is used as an input or control at the destination node. Each output description is referenced to a single IDEF diagram number as indicated in the output map.

Table 6 provides a list of the variables used in the model. The list provides a glossary of the following three kinds of symbols: index variables, arguments of functions, and variables.

Table 3. IDEF Node List

Node	Title	IDEF Nr.
A-O	Optimization of Simulation-Based Training Systems	1
AO	Optimize Training System Design	2
A1	Perform Preliminary Processing	3
A11	Analyze Tasks for Instructional Features	
A12	Analyze Tasks for Fidelity Features	
A13	Determine Task Weights	4
A131	Compute Investment Allocation	
A132	Compute Fixed Operating Allocation	
A133	Compute Variable Operating Allocation	
A134	Compute Normalize Economic Weights	
A14	Determine Learning Function	5
A141	Determine Transfer Ratio Parameters	6
A1411	Determine Benchmark Transfer	
A1412	Determine Equation Exponent	
A1413	Determine Reference Point	
A1414	Determine Equation Multiplier	
A142	Determine Common Learning Curve Parameters	7
A1421	Determine Scaling Factors	
A1422	Determine AE Headstart	
A143	Determine Specific Learning Curve Parameter	8
A1431	Determine Transfer Ratios	
A1432	Determine Time Multiplier	9
A14321	Determine Instructional Feature Improve	
A14322	Determine AE Useful Time Ratio	
A14323	Determine Device Useful Time Ratio	
A14324	Calculate Time Multiplier	
A1433	Determine Head Start	
A2	Develop Training Concept	10
A21	Recommend Simulator Configuration	11
A211	Evaluate Each Task	12
A2111	Evaluate Simulation Indicators	13
A21111	Rate Simulation Requirements	
A21112	Calculate Operating Cost Savings	14
A211121	Minimize Total Operating Cost	
A211122	Calculate Cost Savings	
A211123	Calculate Index of OCS	
A21113	Calculate Simulation Indicators	
A2112	Estimate Development Costs	15
A21121	Calculate Fidelity Dimension Costs	
A21122	Sum Across Dimensions	
A212	Develop Simulation Recommendations	16
A2121	Prepare Scatterplot of Tasks	
A2122	Calculate Simulator Option Summaries	17
A21221	Compute Cost Boundary	
A21222	Assign Tasks to FMS/PMS/AE Options	
A21223	Sum Task Weights for FMS/PMS/AE Options	

Table 3 (continued). IDEF Node List

Node	Title	IDEF Nr.
A3	Design Training Devices	18
A31	Select Instructional Features	19
A311	Select Tasks and Candidate Features	20
A3111	Select Device Tasks	
A3112	Select Instructional Features	
A312	Calculate Benefits of Features	21
A3121	Calculate Effectiveness Score	
A3122	Calculate Feature Benefit	
A313	Select Optimal Features	22
A3131	Compute Benefit/Cost Ratios for Options	
A3132	Sort Options by Benefit/Cost Ratios	
A3133	List Optimal Features	
A32	Optimize Device Fidelity	23
A321	Construct Training Device Options	24
A3211	Select Tasks and Relevant Dimensions	
A3212	Set Minimum and Maximum Options	
A3213	Incorporate Instructional Features Options	
A3214	Define Intermediate Options	
A322	Calculate Costs and Benefits of Options	25
A3221	Calculate Costs	
A3222	Determine Task Trainability	
A3223	Calculate Benefit Scores	
A3224	Calculate Benefit Weights	
A323	Compute Optimal Device Designs	26
A3231	Compute Benefit/Cost Ratios for Options	
A3232	Sort Options by Benefit/Cost Ratios	
A3233	Compute Device Designs	
A3234	Calculate Cost Elements	
A4	Assign Training to Devices	27
A41	Select Training Device for Tasks	28
A411	Determine Training Device Hourly Cost	
A412	Determine Cost Effectiveness Devices	29
A4121	Find First Device	
A4122	Set Current Performance Level	
A4123	Find Device Crossover Points	
A4124	Process Crossover Points	
A4125	Calculate Training Time & Costs	
A413	Examine Device Utilization	30
A4131	Calculate Total Device Hours	
A4132	Calculate Number of Simulators Required	
A4133	Determine Actual Utilization	
A4134	Update Assumption	

Table 3 (continued). IDEF Node List

Node	Title	IDEF Nr.
A42	Allocate Trng Resources to Trng Devices	31
A421	Detail Cost Curves	32
A4211	Detail Cost Curves, Within Control Window	33
A42111	Construct Base Two-Segment Cost Curve	
A42112	Add Next Segment at Current Usage	
A42113	Add New Left Segment	
A42114	Add New Right Segment	
A42115	Check for Completion and Terminate	
A4212	Simplify Using Control Window Shrinkage	
A4213	Simplify Using Detail-Sensitivity Ordering	
A4222	Solve Multi-Task Resource Allocation Problem	34
A4221	Rank Cost Curve Combinations by Minimum Potential Costs	
A4222	Determine Multi-Task Resource Allocation Specific Cost Curve Combination	35
A42221	Conduct Checks and Initializations for Current Combination	
A42222	Set Up and Solve Single-Task Subproblems	36
A422221	Begin Next Algorithm Pass	
A422222	Determine Device for Next Step in Backward Allocation	
A422223	Determine Next Switching Performance Level	
A422224	Allocate Resources to Step, Forcing Lumping from Previous Pass	
A422225	Determine Next Algorithm Action	
A4223	Characterize Solution and Adjust Surcharges To Enforce Constraint Compliance	37
A42231	Check for Constraint Violation and Adjust Surcharges	
A42232	Check for Satisfactory Convergence and Check for Optimality of Combination	

Table 4. Data Base Map

DB Cat.	Description	Destination	Input or Control	IDEF Number
1	Task Training Requirements	AO	Control	OPT-1
		A1	Control	OPT-2
		A2	Control	OPT-2
		A3	Control	OPT-2
		A4	Control	OPT-2
		A21	Control	OPT-10
		A31	Control	OPT-18
		A311	Control	OPT-19
		A3111	Control	OPT-20
		A41	Control	OPT-27
		A42	Control	OPT-27
1A	Task Learning Points	A11	Input	OPT-3
		A14	Input	OPT-3
		A142	Input	OPT-5
		A211	Control	OPT-11
		A2111	Control	OPT-12
		A21112	Control	OPT-13
		A211121	Control	OPT-14
1A1	Entry Performance Level	A1421	Input	OPT-7
		A1422	Input	OPT-7
		A1433	Input	OPT-8
		A412	Control	OPT-28
		A4121	Control	OPT-29
		A4122	Control	OPT-29
		A422	Control	OPT-31
		A4222	Control	OPT-34
		A42222	Control	OPT-35
		A422222	Control	OPT-36
		A422223	Control	OPT-36
		A422224	Control	OPT-36
1A2	Performance Standard	A141	Input	OPT-5
		A1411	Input	OPT-6
		A1421	Input	OPT-7
		A412	Control	OPT-28
		A4124	Control	OPT-29
		A422	Control	OPT-31
		A4222	Control	OPT-34
		A42222	Control	OPT-35
		A422221	Control	OPT-36
1B	Simulation Requirements	A211	Control	OPT-11
		A2111	Control	OPT-12
		A21111	Control	OPT-13

Table 4 (continued). Data Base Map

DB Cat.	Description	Destination	Input or Control	IDEF Number
2	Other Task Data	AO	Input	OPT-1
		A1	Input	OPT-2
		A2	Input	OPT-2
2A	Task Training Hours and Costs	A13	Input	OPT-3
		A14	Input	OPT-3
		A131	Input	OPT-4
		A133	Input	OPT-4
		A142	Input	OPT-5
		A143	Input	OPT-5
		A1432	Input	OPT-8
		A14322	Input	OPT-9
		A14323	Input	OPT-9
		A21	Input	OPT-10
		A211	Input	OPT-11
		A2111	Input	OPT-12
		A21112	Input	OPT-13
		A211121	Input	OPT-14
2A5	Other training costs	A123	Input	OPT-4
2B	Task Information Processing Characteristics	A11	Input	OPT-3
		A12	Input	OPT-3
2C	Task Activities	A12	Input	OPT-3
3	Training Device Data	AO	Input	OPT-1
		A1	Input	OPT-2
		A2	Input	OPT-2
		A4	Input	OPT-2
		A21	Input	OPT-10
		A211	Input	OPT-11
		A212	Input	OPT-11
		A42	Input	OPT-27
3A	Training Device Costs	A13	Input	OPT-3
		A2111	Input	OPT-13
		A21112	Input	OPT-13
		A211121	Input	OPT-14
		A211122	Input	OPT-14
		A41	Input	OPT-27
		A411	Input	OPT-28
		A421	Input	OPT-31
		A4211	Input	OPT-32
		A422	Input	OPT-32
		A42111	Input	OPT-33
3A1	Investment Cost	A131	Input	OPT-4

Table 4 (continued). Data Base Map

DB Cat.	Description	Destination	Input or Control	IDEF Number
3A2	Annual Fixed Operating Cost	A132	Input	OPT-4
3A3	Hourly Variable Operating Cost	A133	Input	OPT-4
3A4	Lifecycle	A13	Control	OPT-3
		A131	Control	OPT-4
		A2122	Input	OPT-16
		A21221	Input	OPT-17
3A5	Maximum Annual Utilization	A131	Input	OPT-4
		A4132	Control	OPT-30
3B	Training Device Cue & Response Capabilities	A14	Input	OPT-3
		A143	Input	OPT-5
		A1431	Input	OPT-8
3C	Training-Device Instructional Features	A14	Input	OPT-3
		A143	Input	OPT-5
		A1432	Input	OPT-8
		A14321	Input	OPT-9
4	Fidelity Dimension Data	AO	Input	OPT-1
		A2	Input	OPT-2
		A3	Input	OPT-2
		A21	Input	OPT-10
		A211	Input	OPT-11
		A32	Input	OPT-18
4A	Fidelity Dimensions and Levels	A2112	Input	OPT-12
		A21121	Input	OPT-15
		A321	Input	OPT-23
		A3211	Input	OPT-24
		A3212	Input	OPT-24
		A3214	Input	OPT-24
4B	Fidelity Dimension Cost Data	A2112	Input	OPT-12
		A21121	Input	OPT-15
		A322	Input	OPT-23
		A3221	Input	OPT-25
4C	Minimum Performance Parameter	A14	Input	OPT-3
		A141	Input	OPT-5
		A1412	Input	OPT-6
		A1413	Input	OPT-6
4D	Fidelity Rules	AO	Control	OPT-1
		A1	Control	OPT-2
		A12	Control	OPT-3

Table 4 (continued). Data Base Map

DB Cat.	Description	Destination	Input or Control	IDEF Number
5A	Instructional Feature Rules	AO	Control	OPT-1
		A1	Control	OPT-2
		A11	Control	OPT-3
5B	Instructional Feature Cost and Weight	AO	Input	OPT-1
		A3	Input	OPT-2
		A31	Input	OPT-18
		A311	Input	OPT-19
		A3112	Input	OPT-20
5B1	Instructional Feature Cost	A313	Input	OPT-19
		A3131	Input	OPT-22
5B2	Instructional Feature Benefit Weight	A312	Input	OPT-19
		A3122	Input	OPT-21
6	Training System Data	AO	Control	OPT-1
		A1	Control	OPT-2
		A2	Control	OPT-2
		A3	Control	OPT-2
		A4	Control	OPT-2
		A22	Control	OPT-10
		A211	Control	OPT-11
		A212	Control	OPT-11
		A2111	Control	OPT-12
		A21113	Control	OPT-13
		A2122	Control	OPT-16
		A41	Control	OPT-27
		A42	Control	OPT-27
		A413	Control	OPT-28
6A1	Annual Student Throughput	A13	Control	OPT-3
		A131	Control	OPT-4
		A4132	Control	OPT-30
		A4133	Control	OPT-30
6B	Model Information	A14	Control	OPT-3
		A143	Control	OPT-5
		A1432	Control	OPT-8
		A32	Control	OPT-18
		A323	Control	OPT-23
		A3234	Control	OPT-26
6B1	Standard Adjustment Rate	A141	Control	OPT-5
		A1411	Control	OPT-6

Table 4 (continued). Data Base Map

DB Cat.	Description	Destination	Input or Control	IDEF Number
6B2	Learning Curve Exponent	A142	Control	OPT-5
		A143	Control	OPT-5
		A1421	Control	OPT-7
		A1422	Control	OPT-7
		A1433	Control	OPT-8
6B3	Maximum Instructional Feature Effect	A14321	Control	OPT-9
6B5	Assumed Setup Savings Percentage	A13323	Control	OPT-9
6B6	Cost Savings Weight	A21221	Control	OPT-17
6B7	Recommendation Boundaries	A21222	Control	OPT-17

Table 5. IDEF Output Map

Output Cat.	Description	Source	Destination	Input or Control	IDEF Num.
O1A	Instructional Feature Appropriateness	A1	A3	Input	2
		A11	A14	Control	3
		A11	UP		3
O1B	Task Cue & Response Requirements	A1	A2	Control	2
		A1	A3	Control	2
		A12	A14	Control	3
		A12	UP		3
O1C	Task Economic Weights	A1	A2	Input	2
		A1	A3	Input	2
		A13	UP		3
		A134	UP		4
O1D	Learning Function Parameters	A1	A2	Input	2
		A1	A4	Input	2
		A14	UP		3
O1D1	Time Scaling Factor	A142	UP		5
		A1421	A1422	Control	7
		A1421	UP		7
O1D2	AE Head Start	A142	A143	Control	5
		A142	UP		5
		A1422	UP		7
O1D3	Device Specific Learning Parameters	A143	UP		5
O1D3A	Learning Curve Asymptotes	A1431	A1433	Control	8
		A1431	UP		8
O1D3B	Time Multiplier	A1432	A1433	Control	8
		A1432	UP		8
		A14324	UP		9
O1D3C	Head Start	A1433	UP		8
O1E	Baseline Variable Operating Costs	A1	A2	Input	2
		A13	UP		3
		A133	UP		4
		A133	A134	Input	4
O13A	Total AE Training Time	A13	A14	Input	3
		A133	UP		4
O131A	CINV	A131	A134	Input	4
O131B	TAE	A131	A133	Input	4

Table 5. (continued). IDEF Output Map

Output Cat.	Description	Source	Destin-ation	Input or Control	IDEF Num.
0131C	NAE & TAENORM	A131	A132	Input	4
0132A	CFX	A132	A134	Input	4
0141A	Intermediate Parameters	A141	A143	Control	5
0141A1	AR	A1411	A1412	Control	6
		A1411	UP		6
0141A2	Q	A1412	A1413	Control	6
		A1412	UP		6
0141A3	b	A1413	A1414	Control	6
		A1413	UP		6
0141A4	q	A1414	UP		6
014321A	IADJ T,i	A14321	A14324	Input	9
014322A	U T,AE	A14322	A14324	Input	9
014323A	U T,i	A14323	A14324	Input	9
O2A	Simulation Recommendation	A2	A3	Control	2
		A2	UP		2
		A21	UP		10
		A212	UP		11
O2A1	Task Assignments to Options	A2122	UP		16
		A21222	A21223	Control	17
		A21222	UP		17
O2A2	Option Summaries	A2122	UP		16
		A21223	UP		17
O211A	Simulation Requirements Indicators	A211	A212	Input	11
		A2111	UP		12
		A21113	UP		13
O211B	Development Cost Estimates	A211	A212	Input	11
		A2112	UP		12
		A21122	UP		15
O21111A	Simulation Requirements Indices	A21111	A21113	Input	13
O21112A	Operating Cost Savings	A21112	UP		13
		A211122	A211123	Input	14
		A211122	UP		14

Table 5. (continued). IDEF Output Map

Output Cat.	Description	Source	Destin-ation	Input or Control	IDEF Num.
O21112B	Operating Cost Savings Indices	A21112 A211123	A21113 UP	Input	13 14
O211121A	AE Hours Savings	A211121 A211121	A211122 A211123	Control Control	14 14
O211121B	FMS Training Time	A211121	A211122	Control	14
O21121A	Required Cue & Response Cost	A21121	A21122	Input	15
O2121A	Scatterplot	A2121	A2122	Control	16
O21221A	AE Cost Boundary	A21221	A21222	Control	17
O3A	Optimal Device Designs	A3	UP		2
O3A1	Selected Instructional Features	A31 A31 A313 A3133	A32 UP UP UP	Input	18 18 19 22
O3A2	Optimal Device Capability	A3 A32 A323 A3233	A1 UP UP UP	Input	2 18 23 26
O3A3	Device Cost Elements	A3 A32 A323 A3234	A4 UP UP UP	Input	2 18 23 26
O311A	Candidate Instructional Features	A311 A311 A3112	A312 A313 UP	Control Control	19 19 20
O311B	Selected Tasks	A311 A3111 A3111	A312 A3112 UP	Control Control	19 20 20
O312A	Feature Benefit	A312 A3122	A313 UP	Input	19 21
O3121A	Weighted Feature Effectiveness	A3121	A3122	Input	21
O3131A	Benefit/Cost Ratio for All Options	A3131	A3132	Input	22

Table 5. (continued). IDEF Output Map

Output Cat.	Description	Source	Destin-ation	Input or Control	IDEF Num.
O3132A	Benefit/Cost Priority of Options	A3132	A3133	Input	22
O321A	Training Device Options	A321	A322	Control	23
		A321	A323	Control	23
		A3214	UP		24
O321B	Technical Performance Indices	A321	A322	Input	23
		A3214	UP		24
O3211A	Relevant Tasks	A3211	A3212	Control	24
O3211B	Relevant Dimensions	A3211	A3212	Input	24
		A3211	A3214	Input	24
O3212A	Minimum and Maximum Options	A3212	A3214	Control	24
O3213A	Instructional Feature Options	A3213	A3214	Control	24
O322A	Costs and Benefits of Options	A322	A323	Input	23
O322A1	Option Costs	A3221	UP		25
O322A2	Option Benefit Scores	A3223	A3224	Input	25
		A3223	UP		25
O322A3	Fidelity Dimension Weights	A3224	UP		25
O3222A	Tasks Trained by Option	A3222	A3223	Input	25
O3231A	Benefit/Cost Ratios For Options	A3231	A3232	Input	26
O3232A	Benefit/Cost Priority of Options	A3232	A3233	Input	26
O3233A	Selected Design Development Cost	A3233	A3234	Input	26
O4A	Efficient Devices	A4	A3	Control	2
		A4	UP		2
		A41	A42	Input	27
		A41	UP		27

Table 5. (continued). IDEF Output Map

Output Cat.	Description	Source	Destination	Input or Control	IDEF Num.
O4A1	Selected Devices	A412	UP		28
		A4121	A4123	Input	29
		A4121	UP		29
		A4124	UP		29
O4A2	Costs	A412	UP		28
		A4125	UP		29
O4A3	Hours/Task	A412	A413	Input	28
		A412	UP		28
		A4125	UP		29
O4B	Optimal Allocation	A4	UP		2
		A42	UP		27
		A423	UP		31
O411A	Device Hourly Cost	A411	A412	Control	28
O411B	Currently Assumed Utilization	A411	A413	Input	28
O4122A	Current Performance Level	A4122	A4123	Input	29
		A4122	A4124	Control	29
		A4122	A4125	Input	29
O4123A	Crossover Points	A4123	A4124	Control	29
O4124A	Device Performance Criterion	A4124	A4122	Input	29
		A4124	A4125	Input	29
O413A	Revised Device Utilization	A413	A411	Input	28
		A4134	UP		30
O4131A	Total Hours by Device	A4131	A4132	Input	30
		A4131	A4133	Input	30
O4132A	Required Training Devices	A4132	A4133	Control	30
O4133A	Actual Utilization	A4133	A4134	Control	30
O421A	Number of Combinations	A421	A422	Input	31
O421B	Cost Curve Parameters	A421	A422	Input	31
		A421	A423	Input	31
		A4213	UP		32
O4211A	Candidate Cost Curve Parameters	A4211	A4212	Input	32
		A42115	UP		33

Table 5. (continued). IDEF Output Map

Output Cat.	Description	Source	Destin-ation	Input or Control	IDEF Num.
O42111A	Interim Cost Curve Parameter	A42111	A42112	Input	33
O42112A	Interim Cost Curve Parameter	A42112	A42113	Input	33
O42113A	Interim Cost Curve Parameter	A42113	A42114	Input	33
O42113B	NEWL, NEWR, NSEGL, NSEGR	A42113	A42113	Control	33
O42114A	Interim Cost Curve Parameter	A42114	A42115	Input	33
O42114B	NEWL, NEWR, NSEGL, NSEGR	A42114	A42113	Control	33
O42115A	Program Control	A42115	A42113	Control	33
O42115B	NEWL, NEWR, NSEGL, NSEGR	A42115	A42113	Control	33
O4212A	Revised Control Window Parameter	A4212	A4211	Control	32
O4212B	Revised Cost Curve Parameters	A4212 A4212	A4213 UP	Input	32 32
O4213A	Identity of Least Detail Sensitive Device	A4213	A4211	Control	32
O422A	Current Resource Allocation Solution	A422 A4223 A4223	A423 A4222 UP	Control Control	31 34 34
O422A1	Incumbent Best Combo and Solution	A42232	UP		37
O4221A	Cost Combo Ranking	A4221	A4222	Control	34
O4221B	Minimum Potential Costs	A4221	A4222	Input	34
O4222A	Tentative Resource Allocation Solution	A4222 A42222	A4223 UP	Input	34 35
O4222A1	Selected Device Training Sequence	A422225	UP		36
O4222A2	Selected Device Resource Allocation	A422225	UP		36
O42221A	Initial Surcharges	A42221	A42222	Input	35

Table 5. (continued). IDEF Output Map

Output Cat.	Description	Source	Destination	Input or Control	IDEF Num.
O42221B	Resource Allocation Upper Bounds	A42221	A42222	Input	35
O42222A	Training Task Counter	A42222	A42222	Control	35
		A422225	A422221	Control	36
O422221A	Initial Number of Stages	A422221	A422222	Input	36
O422221B	Current Performance Level	A422221	A422222	Input	36
O422221C	Effective Device Cost Rates	A422221	A422222	Input	36
		A422221	A422223	Input	36
O422222A	Incumbent Device	A422222	A422223	Input	36
O422222B	Current Performance Level	A422222	A422223	Input	36
O422223A	Next Device or No-Next Device	A422223	A422225	Input	36
O422223B	Next Performance Level	A422223	A422224	Input	36
O422224A	Number of Stages	A422224	A422225	Input	36
O422224B	Multiple Allocation Flags	A422224	A422225	Input	36
O422224C	Current Performance Level	A422224	A422225	Input	36
O422224D	Aggregate Allocation to Incumbent Device	A422224	A422225	Input	36
O422225A	Pass	A422225	A422221	Control	36
O422225B	Incumbent Device	A422225	A422223	Input	36
O422225C	Current Performance Level	A422225	A422222	Input	36
		A422225	A422223	Input	36
O4223A	Iteration Counter	A4223	A4222	Control	34
		A42232	UP		37
O4223B	Surcharges	A4223	A4222	Control	34
		A42231	A42232	Input	37
		A42231	UP		37

Table 5. (continued). IDEF Output Map

Output Cat.	Description	Source	Destin-ation	Input or Control	IDEF Num.
O4223C	Cost Curve Combo	A4223 A42232	A4222 UP	Control	34 37
O423A	Iteration and Flag Counter	A423	A421	Control	31
O423B	Last Resource Solution	A423	A421	Control	31

Table 6. OSBATS Variable List

Indices:

AE = Actual equipment as a training device
FMS = Full mission simulator
i = Index of training devices
INCMB = Index of the incumbent training device
INEXT = Index of the next device to be used for training
j = Index of cue/response dimensions
k = Index of instructional features
KB = Index of ranks
KBEST = Index of best solution
KOMBO = Index of cost-curve combinations
m = Index of fidelity dimension options
M = Index of the most capable fidelity dimension option
n = Index of number of devices acquired for cost curve segments
R = Hypothetical training device that exactly meets all cue and response requirements
s = Selected device
T = Index of tasks

Arguments of Functions:

t = Time
TP_{jm} = Absolute technical performance index for training device option m on fidelity dimension j

Variables:

A_{in} = Height of cost curve $CUMC_i(t)$ at beginning of segment n
ALLOCM = Flag to indicate that multiple allocations have occurred on this pass
ASM_{TR} = Benchmark asymptote for task T on hypothetical device R
ASM_{Ti} = Asymptote of learning curve for task T on device i
$ASSETS_i$ = Number of training devices of type i already owned and available for use without investment cost
B_i = Current effective price of device i usage, equal to the product of the current cost curve segment slope, B_{in}, and the current device i surcharge, $SURCHG_i$
B_{in} = Slope of segment n of the cost curve $CUMC_i(t)$
b_{Tj} = Cue and response reference point for task T and fidelity dimension j
BEN_k = Benefit of instructional feature k
BWT_j = Benefit weight for simulator options on cue/response dimension j
$(B/C)_{jm}$ = Incremental, weighted benefit-to-cost ratio of simulator option m on cue/response dimension j

Table 6. OSBATS Variable List

C_{Ti}	=	the cost of using device i on task T, as calculated by Resource Allocation Module single task optimization
CBEST	=	The per-student cost of training under the incumbent best solution
$CDEV_{jm}(TP_{jm})$	=	Development cost of simulator option m with capability (TP_{jm}) on cue/response dimension j
CFX_T	=	Fixed cost per year allocated to the training of task T
$CINV_T$	=	Investment cost per year allocated to the training of task T
CLB_{KOMBO}	=	The lower bound on possible per-student cost is found for combination KOMBO
CLEFT	=	The minimum cost in the range of detailing
$CMAX_j$	=	Development cost of the maximum technical performance index (set to 1.0) on cue/response dimension j
$CMIN_j$	=	Development cost of the minimum technical performance index (not necessarily 0.0) on cue/response dimension j
CONDTN	=	Flag that indicates whether resource allocation solution has been found
CRANGE	=	The range within which a cost curve must be detailed, expressed in dollars per student
$CREL_{jm}(TP_{jm})$	=	Normalized development cost of simulator option m with capability (TP_{jm}) on fidelity dimension j
CRH_T	=	Number of classroom hours needed to train task T in conjunction with the actual equipment from entry to standard without use of a simulator
CRIGHT	=	The maximum cost in the range of detailing
$CUMC_i(t)$	=	Cumulative cost of training on device i as a function of per student training time
$CVAR_T$	=	Variable cost per year allocated to the training of task T
$DCST_{Tj}(FRQT_{Tj})$	=	Development cost needed to achieve the cue/response requirement of task T on dimension j
$DEVC_T$	=	Development cost estimate for a simulator to train task T
$DEVT_i$	=	Total training time across all tasks on device i
$DVIF_{ik}$	=	Presence of instructional feature k on device i
E_j	=	Empirically determined parameter in a development cost equation for cue/response dimension j
EFF_k	=	Aggregation of effectiveness over chosen tasks for instructional feature k
ENT_T	=	Entry level performance for task T
$EPMIN_{Ti}$	=	User-specified minimum entry performance for task T on device i
ERR1	=	The maximum error that can occur if a cost curve curve is simplified to a single infinite segment
ERR2	=	The maximum error that can occur if a cost curve is simplified to exactly two segments

Table 6. OSBATS Variable List

F_m = The set of instructional features included in level m of instructional features dimension in Fidelity Optimization Module

FCA = Fractional change allowed in change of $SURCHG_i$

FCAMAX = Initial setting of FCA

$FCAP_{ij}$ = Cue and response capability on dimension j on device i

FCFAC = Factor for estimating annual fixed cost for training device design from investment cost

$FMIN_j$ = Minimum performance when capability on fidelity dimension j is null but capabilities on all other dimensions are perfect

FOC_i = Fixed operating cost per year of device i

$FRQT_{Tj}$ = Cue and response requirement for task T on fidelity dimension j

GRADS = Required number of graduates per year

HS_{Ti} = Head start for task T on device i

$IADJ_{Ti}$ = Instructional feature improvement in efficiency for task T on device i

$IDEVC_T$ = Development cost index for a simulator to train task T

IDV_k = A sequence of training devices

IFBN = Maximum percentage improvement in training efficiency for instructional features

IFC_k = Development cost for instructional feature k

$IFNM_{Ti}$ = Number of appropriate instructional features for task T on device i

IFT_{Tk} = Instructional features task match for task T and instructional feature k

$IFWT_k$ = Benefit weight for instructional feature k that includes frequency of use, instructor loading and useability

INV_i = Investment cost of one device i

$IOCS_T$ = Index of operations cost savings due to a simulator for task T

$ISIM_T$ = Indicator that a simulator is appropriate for training task T

ISR_T = Index for the importance of simulator requirements for training task T

JC_T = The number of fidelity dimensions, j, such that $FRQT_{Tj} > 0$ and $FMIN_j > 0$ for task T

k_i = The cost curve segment being used for device i under combination KOMBO

$KALLOC_I$ = Counter to keep track of the number of times device I has received allocations on a pass

KASE = A counter that determines whether costs are to be detailed

KBRANK(KB) = the cost-curve combination ranked KB according to the minimum cost

LC_i = Life cycle (in years) of device i

LCEST = Estimate for life cycle (in years) of device designs produced by Fidelity Optimization Module

Table 6. OSBATS Variable List

LEV_i	=	Identification of current segment for cost curve for device i
MAXIF	=	Maximum number of useful instructional features
MAXKB	=	the maximum allowable number of cost-curve combinations
N_i	=	Number of segments in the cost curve, $CUMC_i(t)$
NAE	=	Number of actual equipment devices needed to train the requisite graduates per year without the use of simulators
$NBEN_{jm}$	=	Normalized benefit score of simulator option m on cue/response dimension j
NCOMBO	=	The total number of combinations of cost-curve segments
ND	=	The number of training devices under consideration by the Resource Allocation Module
$NDEV_i$	=	Number of device i's needed to train the requisite number of graduates per year
NEW	=	The number of new devices associated required
NEWL	=	The number of new devices associated with NSEGL
NEWR	=	The number of new devices associated with NSEGR
NOH_T	=	Number of non-operational hours on the actual equipment in order to train task T from entry to standard without use of a simulator
NRANGE	=	The number of segments desired in a cost curve
NSEGL	=	The leftmost deliberately detailed segment of the cost curve
NSEGR	=	The rightmost deliberately detailed segment of the cost curve
NSTAGE	=	Counter of devices selected for task
$NTAB_i$	=	The number of segments in the cost curve for device i
OCS_T	=	Operations-cost savings for task T that result from using a FMS
OH_T	=	Number of operational hours on the actual equipment in order to train task T from entry to standard without use of a simulator
OTC_T	=	Other costs of training task T, such as ammunition
$P_{Ti}(t)$	=	Performance on task T with device i as a function of time
p_{jk}	=	Performance at completion of k-th stage training, on device j_k
P_{MINI}	=	The minimum performance level that can be reached using the incumbent device
P_{next}	=	Transition performance at which device INEXT will be used
P_{NOW}	=	Current performance level
PASS	=	Counter that reflects the current pass through the resource allocation problem
PB	=	Payback period needed for a full mission simulator
PENTRY	=	Performance level at beginning of training on a device

Table 6. OSBATS Variable List

$PERF_i(TRNT_i, P_{now})$ = Performance achieved on training device i after training an amount of time $TRNT_i$, given an entry level of performance P_{now}.

PEXIT = Performance level at conclusion of training on a device

$PINV_i(p)$ = The inverse function of $PERF_i$

$PRATE_i$ = Rate of increase in performance with respect to per-student dollar spent at that performance level.

PX_{si} = Crossover performance level between the selected device and device i

q_T = Multiplier of asymptote equation for task T

Q_T = Exponent of asymptote equation for task T

r = learning curve exponent

$RGRT21_i$ = The increase in possible cost error for device i if a cost curve of a single infinite segment is used instead of a curve with two segments; i.e. the potential regret.

$RGRT31_i$ = The increase in possible cost error for device i if a cost curve of a single infinite segment is used instead of a curve with three or more segments; i.e. the potential regret.

$RGRT32_i$ = The increase in possible cost error for device i if a cost curve of a two segments is used instead of a curve with three or more segments; i.e. the potential regret.

RM_{Tj} = Modified task cue and response requirement for task T and cue/response dimension j, such that it is zero for least capable simulator option on dimension j and one for the most capable option

SADJ = Adjustment % for benchmark transfer

STD_T = Training standard for task T

$STEPHT_i$ = the size of the step between adjacent segments of a detailed cost curve, for device i

SUH_T = Setup hours on the actual equipment for task T

$SURCHG_i$ = The cost surcharge for each device i

SUSAV = Setup time savings

T_n = Starting value of t for segment n of cost curve, $CUMC_i(t)$

TAE_T = Total number of training hours on the actual equipment (not including the classroom) in order to train task T from entry to standard without use of a simulator

$TAENRM_T$ = Normalized value of TAE_T for task T

$TAVAIL_i$ = Current per-student device i time constraint

$TCST_T$ = Total life cycle training cost that can be allocated to task T

T_{in} = Starting value (per-student time, t) for h-th segment of detailed cost curve $CUMC_i(t)$

TLEFT = The lowest time in the in the domain of detailing

TLRNC = Tolerance parameter

TM_{Ti} = Time multiplication factor for task T on device i

Table 6. OSBATS Variable List

TP_{jm} = Absolute technical performance index for training device option m on fidelity dimension j

$TPMAX_j$ = Technical performance of the most capable FMS option for cue/response dimension j

$TPMIN_j$ = Technical performance of the least capable FMS option for cue/response dimension j

$TPREL_{jm}(TP_{jm})$ = Normalized technical performance of simulator option m with capability TP_{jm} on cue/response dimension j

TRANGE = The domain within which a cost curve must be detailed, expressed in hours per student

TRIGHT = The greatest time in the domain of detailing

TRINC = Incremental resources required to reach from P_{now} down to P_{next}

TRN_{Tjm} = Trainability of task T based on its requirement for cue/response dimension j and the cue/response capability of a simulator option m

$TRNT_{Ti}$ = Training time for task T on device i

TSF_T = Time scaling factor for task T

TSIM = number of tasks over which the simulator development cost should be amortized;

$TTOT_T$ = Total number of training hours on the actual equipment (including the classroom) in order to train task T from entry to standard without use of a simulator

TWEAK = Acceleration parameter used in setting $SURCHG_i$

TWT_T = Economic weight for task T

UB_{iT} = Upper bound (constraint) on allocation of device i usage to task T

$U_{T,AE}$ = Useful time ratio for task T on the actual equipment

U_{Ti} = Useful time ratio for task T on device i

$UACT_i$ = Calculated use per year of device i, in hours

UB_{iT} = an upper bound to device i allocations for each task T

$UBEN_{jm}$ = Raw (or non-normalized) benefit score of simulator option m on cue/response dimension j

$UBEN_{jM}$ = Raw benefit score of most capable simulator option on cue/response dimension j

UEST = Estimate for utilization in hours/year for device designed by Fidelity Optimization Module

$UMAX_i$ = Maximum number of hours that device i can be used in a year

$UNOM_i$ = Nominal utilization in hours that device i is used in a year

$USMAX_i$ = User-imposed maximum per-student usage of device i

VCFAC = Factor for estimating hourly variable cost for a training device design from the investment cost

$VHRC_i$ = Variable operating cost (per hour) of device i

$VHRC_{T,AE}$ = Variable cost (per hour) of both the classroom and the actual equipment for task T

Table 6. OSBATS Variable List

$VHRC_{AE(NO)}$	=	Variable cost (per hour) of the actual equipment in the non-operational training mode (assumed for now to be six times $VHRC_{CR}$)
$VHRC_{AE(O)}$	=	Variable cost (per hour) of the actual equipment in the operational training mode
$VHRC_{CR}$	=	Variable cost (per hour) of the classroom
$VHRC_{FMS}$	=	Variable cost (per hour) of the full mission simulator
VIOLTN	=	Flag that indicates a solution that violates constraints
$VTOTC_i$	=	Total variable operating cost (per hour) of device i, including investment and fixed and variable operating costs
WOCS	=	Weight for the importance of operations-cost savings in the computation of $ISIM_T$
WSR	=	Weight for the importance of simulator requirements in the computation of $ISIM_T$
z	=	Intermediate utility variable used in equations

System Description

The description of a single node in the model consists of three components: two diagrams (which may be repeated) and associated explanatory text. The diagram for the node being described is on the right-hand side of the page; its parent is shown on the left-hand side. The text is written beneath the diagrams. If the explanatory description requires more than two pages, both parent and child diagrams are repeated on the next two pages, until the text describing each node is completed. The diagrams describe OSBATS activities from the viewpoint of the model developer. That is, they show the data required by model processes, and describe in detail the relationships presumed by the model.

OPT/A-0: Optimization of Simulation-Based Training Systems (OSBATS)

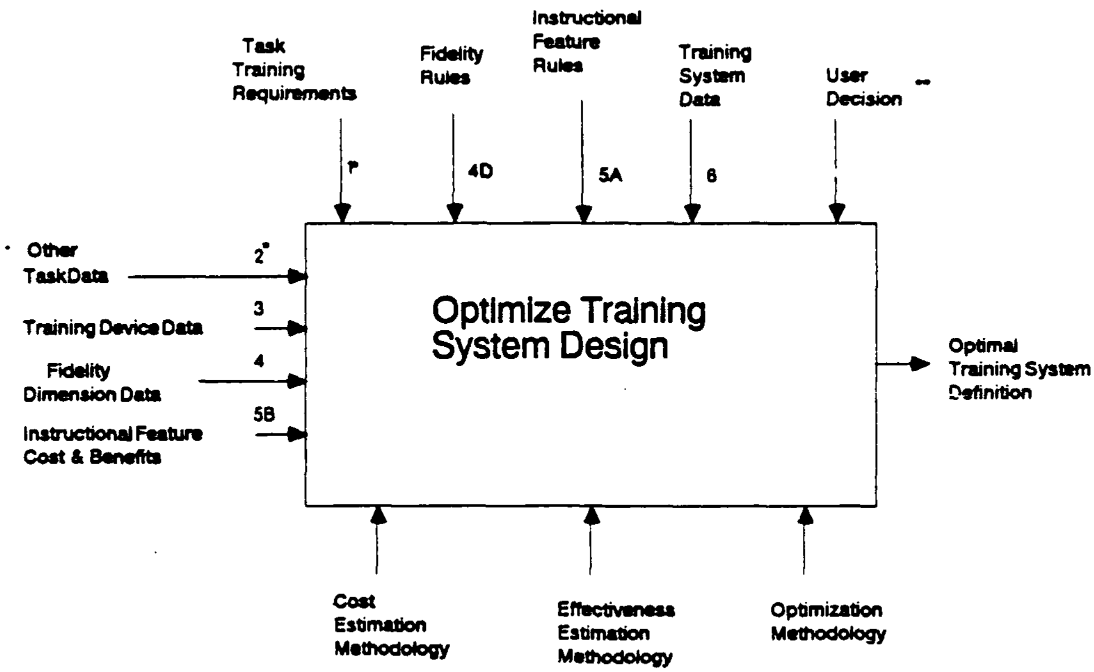
The goal of the OSBATS Model is to specify the designs and concepts of use for training devices to meet training requirements at the least cost, or provide the greatest training effectiveness, given the cost. Training effectiveness is measured by the level of task performance on actual equipment resulting from of the use of the training system. Cost comprises investment, fixed operating, and variable operating components.

The IDEF0 analysis assumes that training requirements are defined by a set of tasks that must be performed to prescribed standards following training. Tasks are rated along several dimensions prior to the application of the OSBATS model. Training devices are characterized by their capability to present cues and collect responses, and by the instructional support features they possess. The population of instructional features and cue/response dimensions is assumed to be known prior to the application of the model, and is represented in the model data base. Training devices evaluated by the OSBATS model may represent currently existing devices, or they may be generated by the design components of the OSBATS model.

The model considers six kinds of input and control data: (1) Task training requirements, (2) Other task data (3) Training device data, (4) Fidelity dimension data, (5) Instructional feature data, and (6) Training system data. Data from each source are used in one or more model component.

Task training requirements describe the tasks to be trained, specifying entry performance level and training standards, safety and special performance conditions, and cue and response requirements.

The two components of other task data describe (1) the training time that would be required to meet the training requirements in one hypothetical case, the case in which all training is conducted in a classroom or on actual equipment, and (2) task information-processing characteristics.



Node: OPT/A-0	Title: Optimization of Simulation-Based Training Systems (OSBATS)	Date: 8/1/88	Number: OPT-1

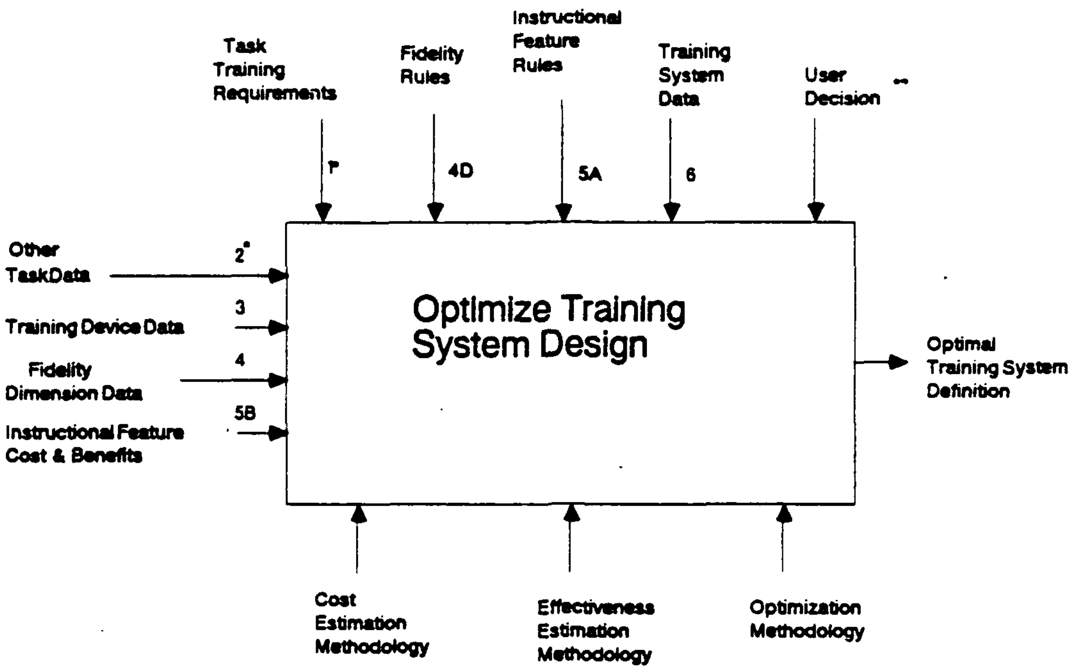
* Labeled inputs and controls refer to elements contained in a database.
** Unlabeled inputs and controls refer to user-supplied information.

Training device data characterize the hypothetical and actual media that might be used to provide training. These data describe the costs, cue and response capabilities, and instructional features present in each alternative device.

Fidelity dimension data characterize the ways that simulators or other training devices may differ in the accuracy with which they present the stimuli and response options from the actual equipment and operating environment in which the tasks are performed. These data are used to design training devices that provide the optimal levels of fidelity on each fidelity dimension as a function of training device cost.

Instructional feature data include the information required to select the instructional features that provide the maximum improvement in training efficiency as a function of cost. This segment of the data base includes rules used to select instructional features, and data used to calculate cost and benefit of instructional features.

Finally, training system data include general information about the training system, and other miscellaneous information required by the model. Specific data include the annual requirement for graduates and several assumptions used by the model.



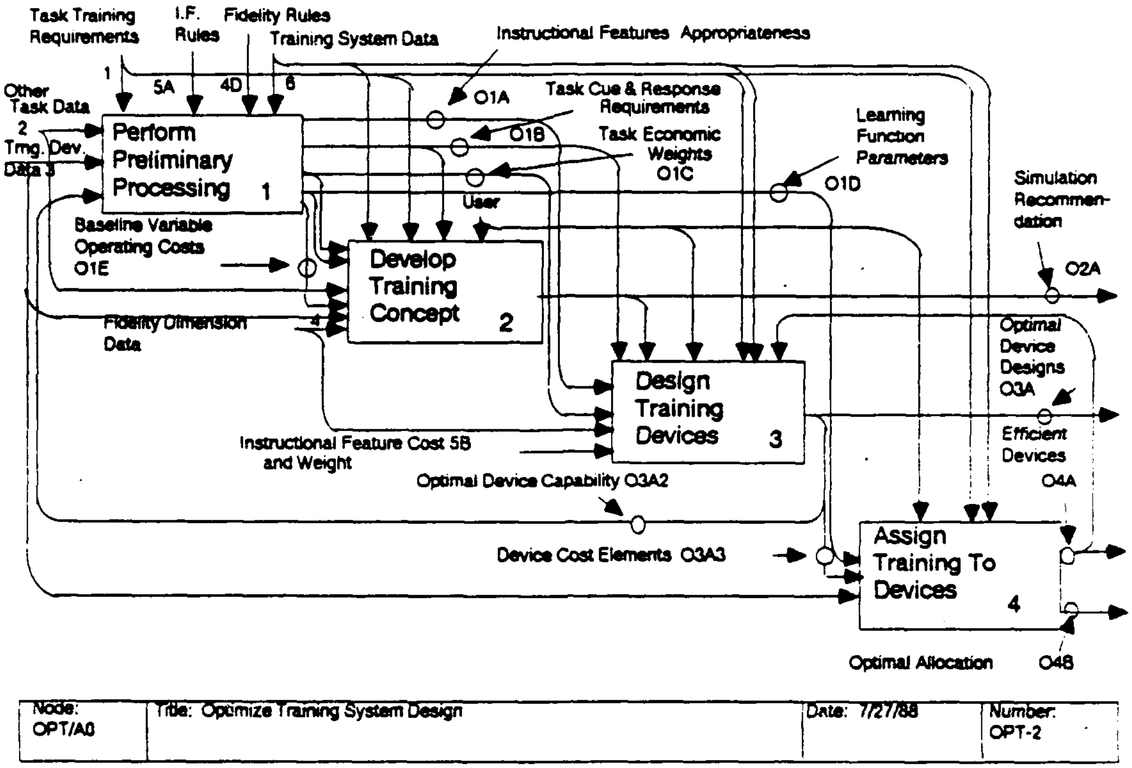
Node: OPT/A-0	Title: Optimization of Simulation-Based Training Systems (OSBATS)	Date: 8/1/88	Number: OPT-1

* Labeled inputs and controls refer to elements contained in a database.
** Unlabeled inputs and controls refer to user-supplied information.

OPT/A0: Optimize Training System Design

The first level of decomposition describes the three problem areas addressed by the model: (1) training-concept development, (2) training device design, and (3) assignment of training resources to devices. Since common processes are involved in several of these problems, the analysis includes a fourth component that conducts the preliminary processing required for more than one model component. The next level of decomposition enumerates modeling tools addressing the problems.

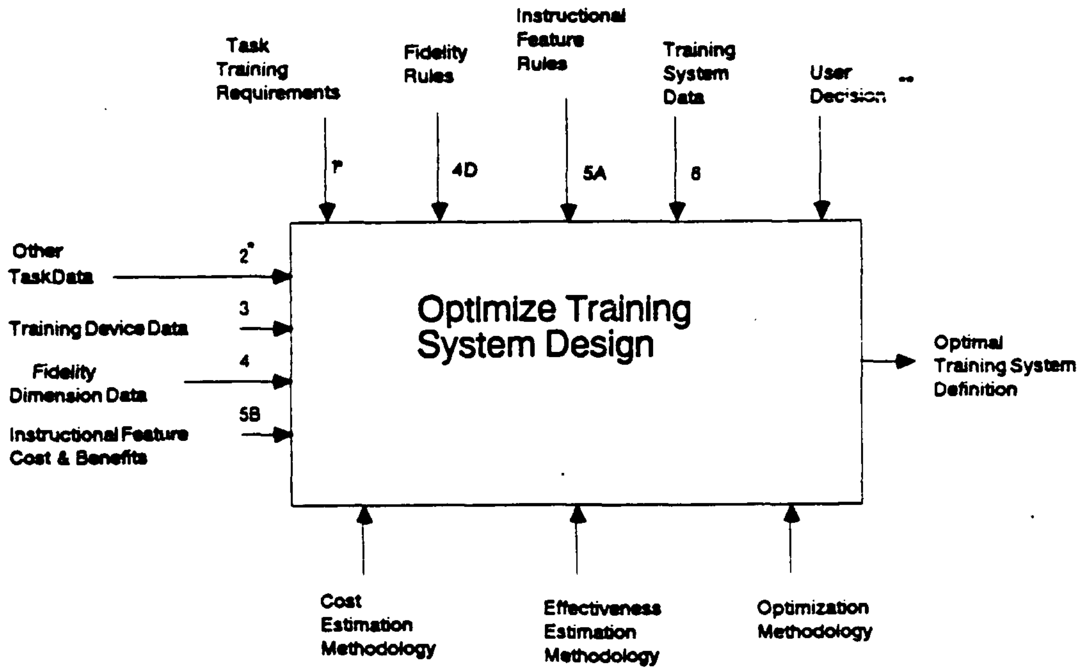
OPT/A1: Perform preliminary processing. This activity produces the following basic information that is processed further by several model components: (1) A task-by-instructional-feature matrix that specifies which instructional features would enhance training for each task, (2) a matrix that describes the cue and response requirements of each task along each fidelity dimension, (3) task weights that reflect the relative cost of training each task on actual equipment, and (4) parameters that describe the course of learning for each task on each candidate training device. Although this activity is preliminary to the operation of any specific tool, the node does receive feedback from the training-device design process in node OPT/A3. That is, devices designed in OPT/A3 will need to be



processed by this node to provide the data required by other modules.

OPT/A2: Develop training concept. This problem area is concerned with finding clusters of tasks that have similar training-device needs. These tasks form training requirements to be used in the training-device design process (OPT/A3). Training-concept development occurs early in the training system development process, and is refined several times during this process. The OSBATS model contains a single tool for training-concept development. This tool evaluates different classes of training devices (full-mission simulators, part-mission simulators, actual equipment) that may meet parts of the training requirements.

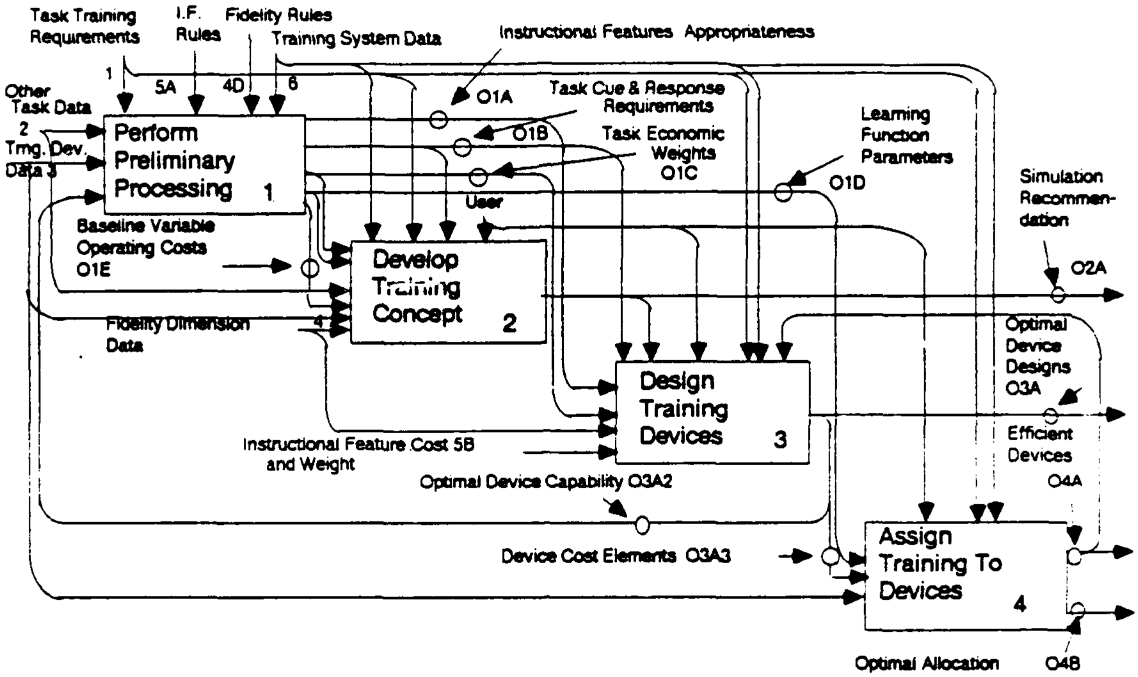
OPT/A3: Design training devices. This problem area is concerned with designing training devices that provide the optimal training effectiveness given the investment cost. The principal problems addressed in the model are (1) providing the optimal level of fidelity with respect to presentation of the environment and equipment, and (2) selecting instructional features that are tailored to the tasks being trained on the training device.



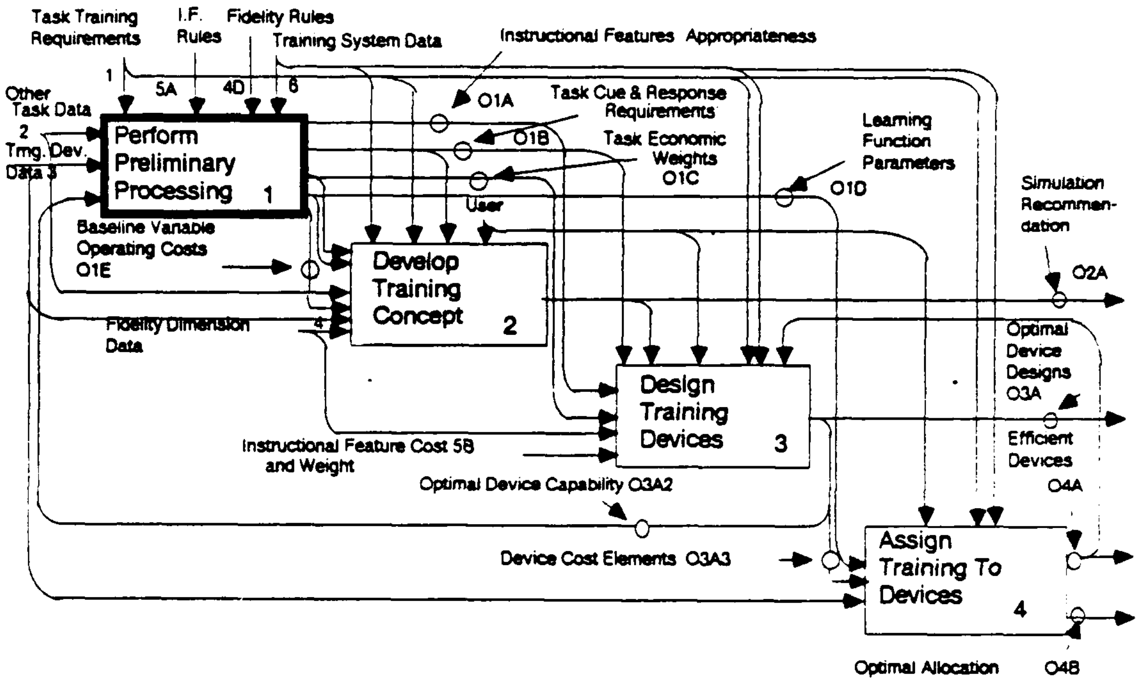
Node: OPT/A-0	Title: Optimization of Simulation-Based Training Systems (OSBATS)	Date: 8/1/88	Number: OPT-1

* *Labeled inputs and controls refer to elements contained in a database.*
** Unlabeled inputs and controls refer to user-supplied information.

<u>OPT/A4: Assign training to devices</u>. This problem is concerned with determining which training device or devices should be used to train each task, and how much training should be conducted on each device. Two tools have been defined for this problem. The training-device selection tool bases the allocation on assumptions that provide for simpler, interactive operation. The resource allocation tool provides a more detailed allocation of training resources to training devices, that considers constraints on training-device use.



Node: OPT/A0	Title: Optimize Training System Design	Date: 7/27/88	Number: OPT-2

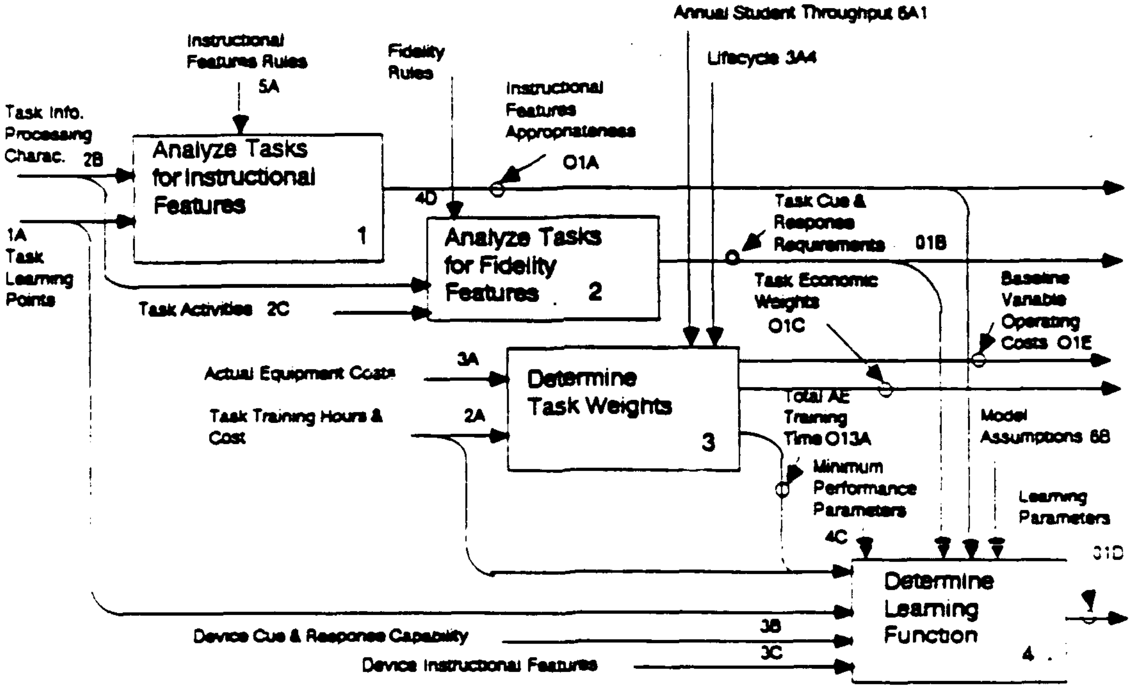


Node: OPT/A0	Title: Optimize Training System Design	Date: 7/27/88	Number: OPT-2

OPT/A1: Perform Preliminary Processing

This activity calculates four sets of variables that are used by other model components. The first subactivity applies a set of rules (5A) that examine characteristics of the tasks to identify effective instructional features. The second subactivity applies a set of rules (4D) to identify task cue and response requirements. The third subactivity determines the total cost to train each task on actual equipment, and normalizes this cost to be used as a task weight. The fourth activity uses task and training-device data to estimate the parameters of training-device learning functions for each task.

OPT/A11: Analyze tasks for instructional features. This activity determines the instructional features that are appropriate for each task. The instructional feature rules (5A) specify the kinds of tasks for which each instructional feature is appropriate. The conditions of these rules are compared to the characteristics of the tasks (2B and 1A) to specify for each task the set of applicable instructional features. The output of this activity is a matrix, $[IFT_{Tk}]$, that indicates whether instructional feature k will enhance the training efficiency of task T. An element IFT_{Tk} of this matrix contains the value 1.0 if instructional feature k is appropriate for task T. This activity is conducted as follows.



Node: OPT/A1	Title: Perform Preliminary Processing	Date: 7/27/88	Number: OPT-3

First, a rule base is interrogated to compare task data and rule conditions. The instructional feature rules (5A) associate instructional features with specific task characteristics. An example of such a rule is given below:

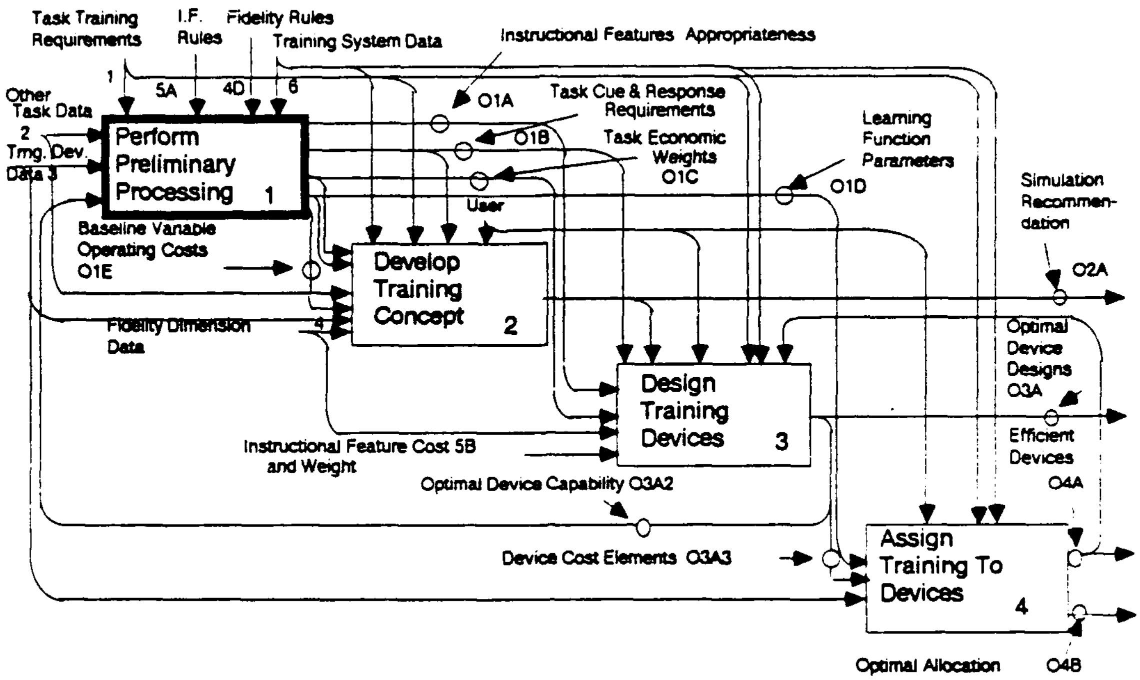
IF: Entry performance (1A1) < 0.4,
and Intrinsic feedback (2B11) is absent,
and The task involves continuous movement (2B1),
or procedures (2B2),
or decision making/rule using (2B4),

THEN: Automated Performance Alerts is indicated for this task.

This activity compares the conditions of the instructional feature rules to the task characteristics (2B) and task learning points (1A), and identifies matches, which it passes on to the next activity.

Next, this activity takes the matches produced in the interrogation, and sets the corresponding cells of IFT_{Tk}. The IFT matrix is defined as follows:

$$IFT_{Tk} = \begin{cases} 1 & \text{if a match was found between feature k and task T} \\ 0 & \text{otherwise.} \end{cases}$$



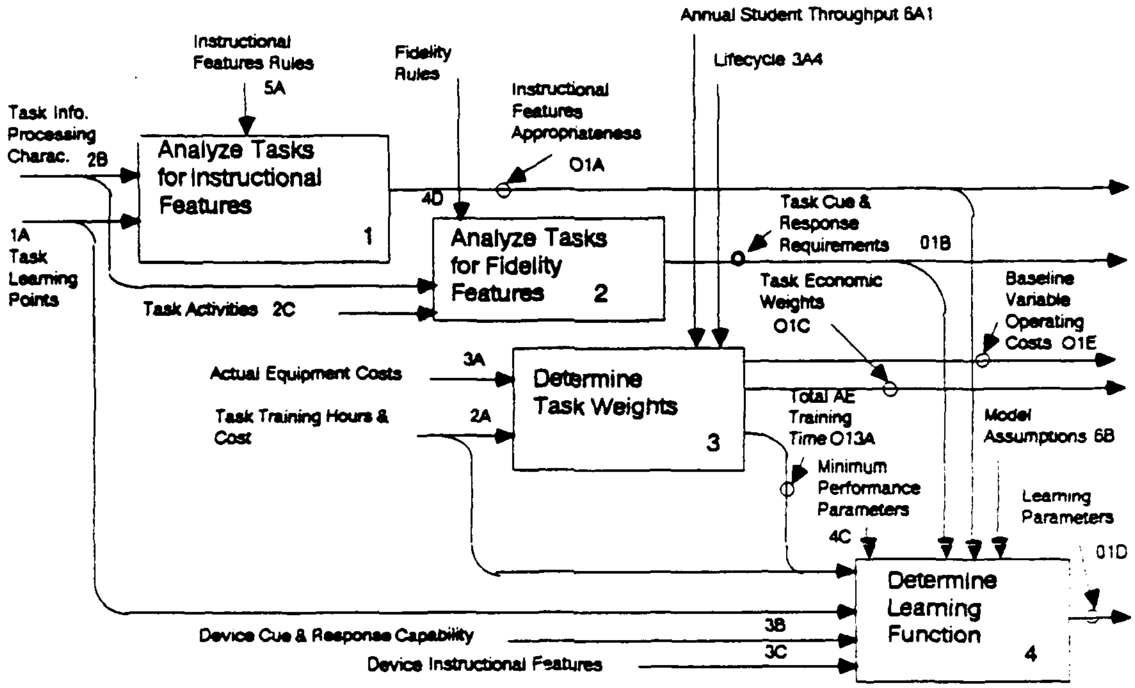
Node: OPT/A0	Title: Optimize Training System Design	Date: 7/27/88	Number: OPT-2

OPT/A12: Analyze tasks for fidelity features. This activity derives the task cue and response requirements ($FRQT_{Tj}$) on each fidelity dimension. The derivation is controlled by the information in the fidelity rule base (4D). The fidelity rule base calculates the $FRQT_{Tj}$ based upon the values of variables that describe task activities (2C) and information processing requirements (2B). The task activity variables are situation and domain dependent. The rule base is hierarchically organized and operates using backward chaining.

The following example shows one of the rules:

IF: Performance cues are provided by longitudinal acceleration,
and Motion cues should be provided by platform motion (as opposed to seat motion),
and The magnitude of the longitudinal acceleration cues is moderate or high,
and The task requires the performance of an emergency procedure,
and The platform longitudinal acceleration provides a cue for the initiation of the emergency procedure

THEN: Platform surge is required for training (the requirement for platform motion is given the value 0.9).



Node: OPT/A1	Title: Perform Preliminary Processing	Date: 7/27/88	Number: OPT-3

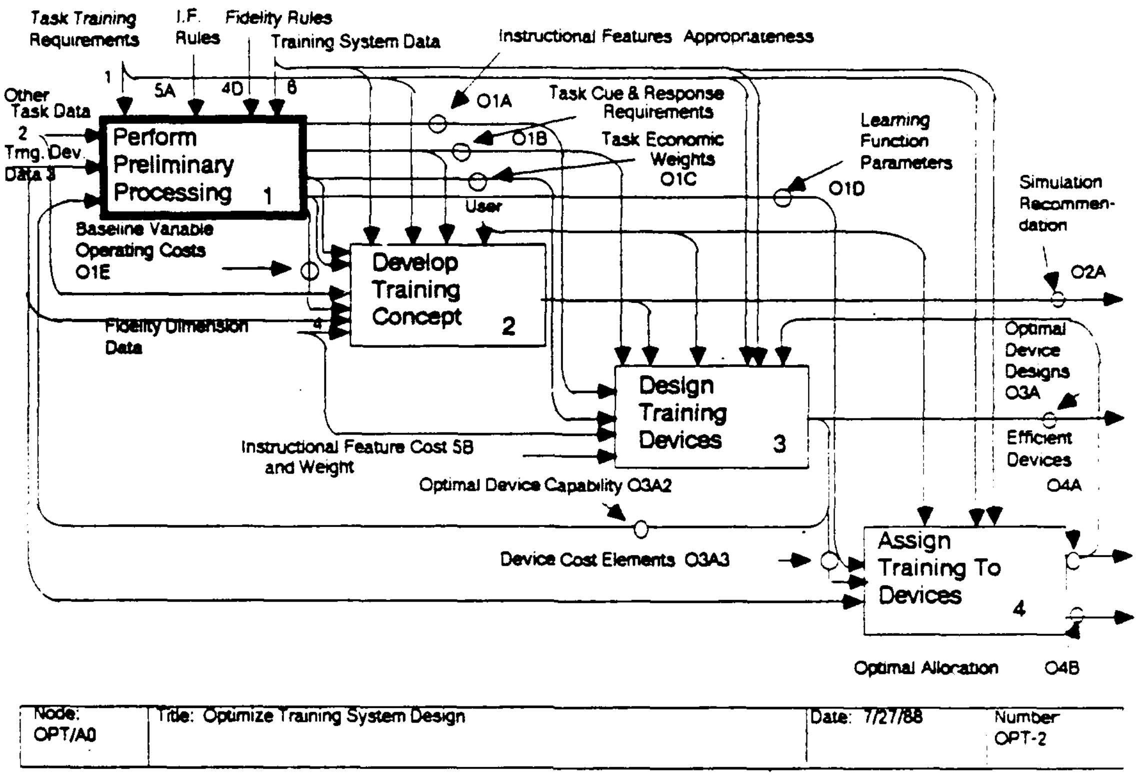
Some of the conditions for this rule are derived from the value of other data; other conditions are direct data values. The resulting values of $FRQT_{Tj}$ are in the interval [0, 1].

<u>OPT/A13: Determine task weights</u>. This activity calculates a set of weights that are used to compare the importance of training improvements for different tasks. The weights are based on the cost of training the tasks on actual equipment. The processes contained in this activity calculate the investment (OPT/A131), fixed operating (OPT/A132), and variable operating (OPT/A133) components of the cost of training on actual equipment. The final process (OPT/A134) combines and normalizes the cost estimates to produce task weights.

<u>OPT/A14: Determine learning function</u>. The basis for the determination of training effectiveness is a learning function that relates performance, $P_{Ti}(t)$, to task and device variables. We use a power function of the following form to describe learning:

$$P_{Ti}(t) = ASM_{Ti}\ \{1 - [1 + TM_{Ti}TSF_T(HS_{Ti} + t)]^{-r}\},$$

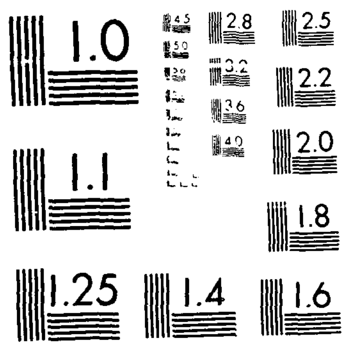
where

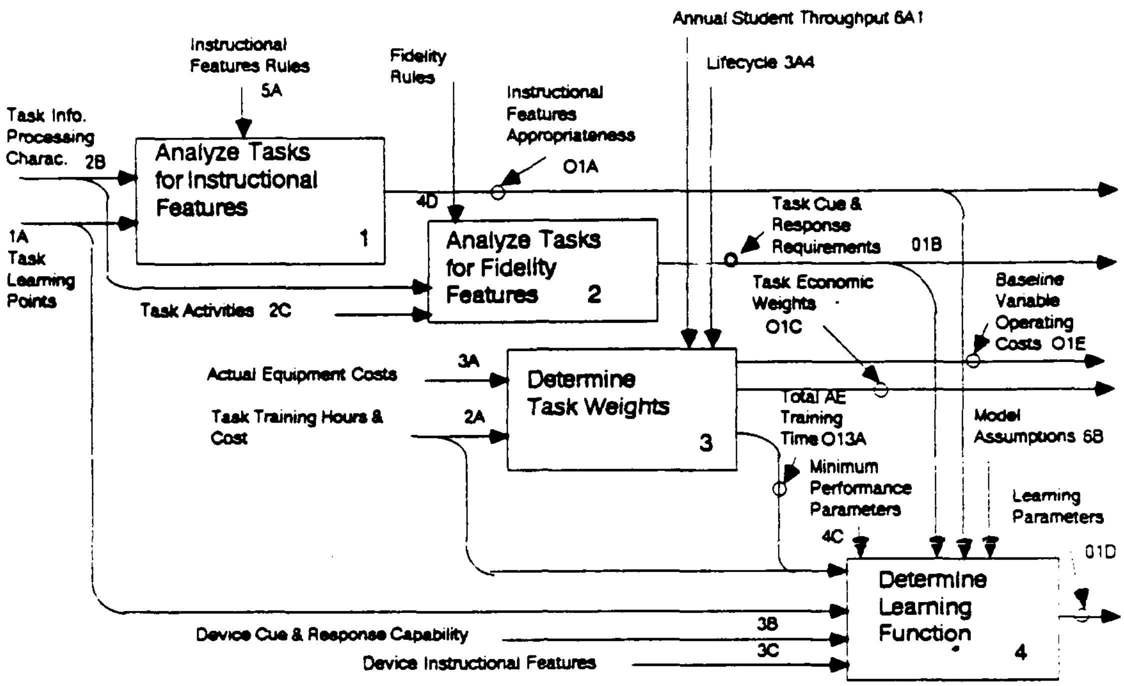


t = the training time,
T = the task to be trained,
i = the training device,
ASM_{Ti} = the asymptote of the learning curve,
TM_{Ti} = the time multiplier for the particular device and task,
TSF_T = the time scaling factor for the task,
HS_{Ti} = the head start for the device and task, and
r = the learning curve exponent.

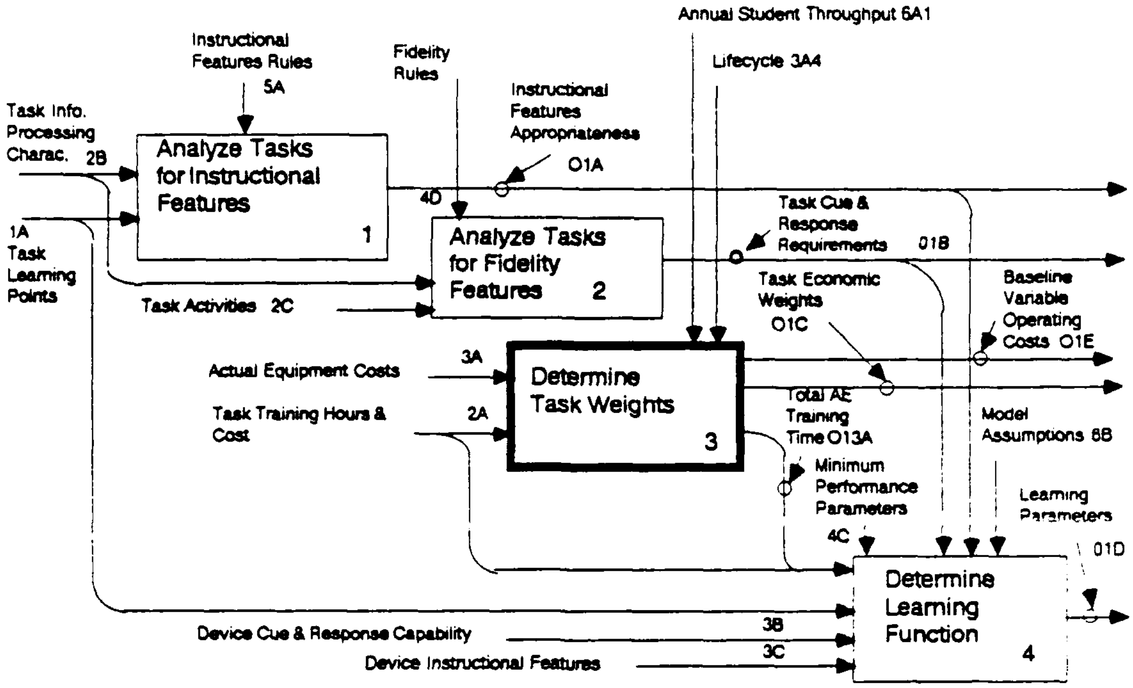
The power function form is used because of the good fit such a function has provided to empirical data (Newell & Rosenbloom, 1981). For actual equipment the parameters of this function may be based on fits to empirical research or on fits to learning curves estimated by subject matter experts (SMEs). For notional equipment a procedure has been developed (OPT/A14) for estimating the learning curve parameters by extrapolation. The processes in this activity first calculate some of the basic parameters needed to calculate the asymptote of the learning function (OPT/A141). Then specific processes estimate the values for the parameters that do not depend on the training device (OPT/A142), and those that do depend on the training device (OPT/A143), respectively.

AD-A237 266 OPTIMIZATION OF SIMULATION-BASED TRAINING SYSTEMS: 2/4
MODEL DESCRIPTION IMPL..(U) HUMAN RESOURCES RESEARCH
ORGANIZATION ALEXANDRIA VA P J STICHA ET AL. JUN 90
UNCLASSIFIED HUMRRO-FR-PRD-88-26 ARI-TR-896 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6



Node: OPT/A1	Title: Perform Preliminary Processing	Date: 7/27/88	Number: OPT-3



Node: OPT/A1	Title: Perform Preliminary Processing	Date: 7/27/88	Number: OPT-3

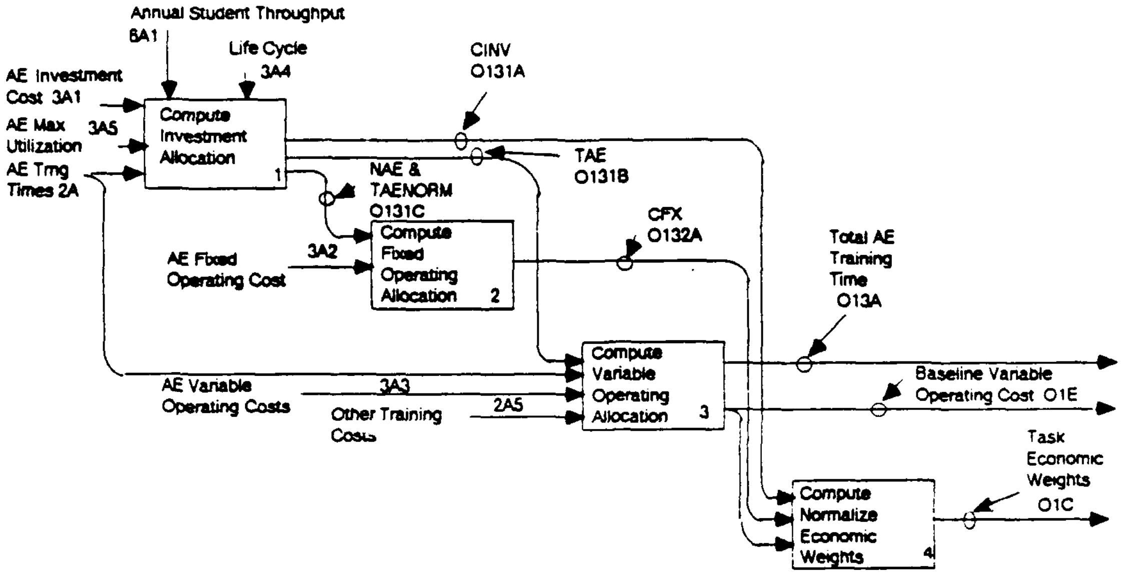
OPT/A13: Determine Task Weights

The task weights are used as a measure of importance of the tasks by several of the modules. They represent the normalized life cycle cost of training the tasks to standard through the use of classroom activities and actual equipment exercises only. The elements of these weights come from the actual equipment training times and other costs (2A), the actual equipment cost data (3A), and training course and system information (6A). The weights are calculated in four steps.

OPT/A131: Compute the investment allocation. The investment in the actual equipment is determined by finding the number of actual equipment devices needed to produce the yearly graduates (GRADS), and then allocating this investment over the equipment life cycle and the tasks using the device in proportion to their usage (classroom investment is ignored). The total training hours for each task on the actual equipment is

$$TAE_T = NOH_T + OH_T + SUH_T$$

where the parameters on the right hand side are data or SME estimates available in the data base:



Node: OPT/A13	Title: Determine Task Weights	Date: 7/27/88	Number: OPT-4

NOH_T = Number of non-operational hours on actual equipment to train task T from entry to standard without a simulator (2A2)

OH_T = Number of operational hours on the actual equipment to train task T from entry to standard without a simulator (2A3)

SUH_T = Setup hours on the actual equipment for task T (2A4).

Since the maximum annual utilization of actual equipment is $UMAX_{AE}$ (3A5), the number of actual equipment devices per year (N_{AE}) is:

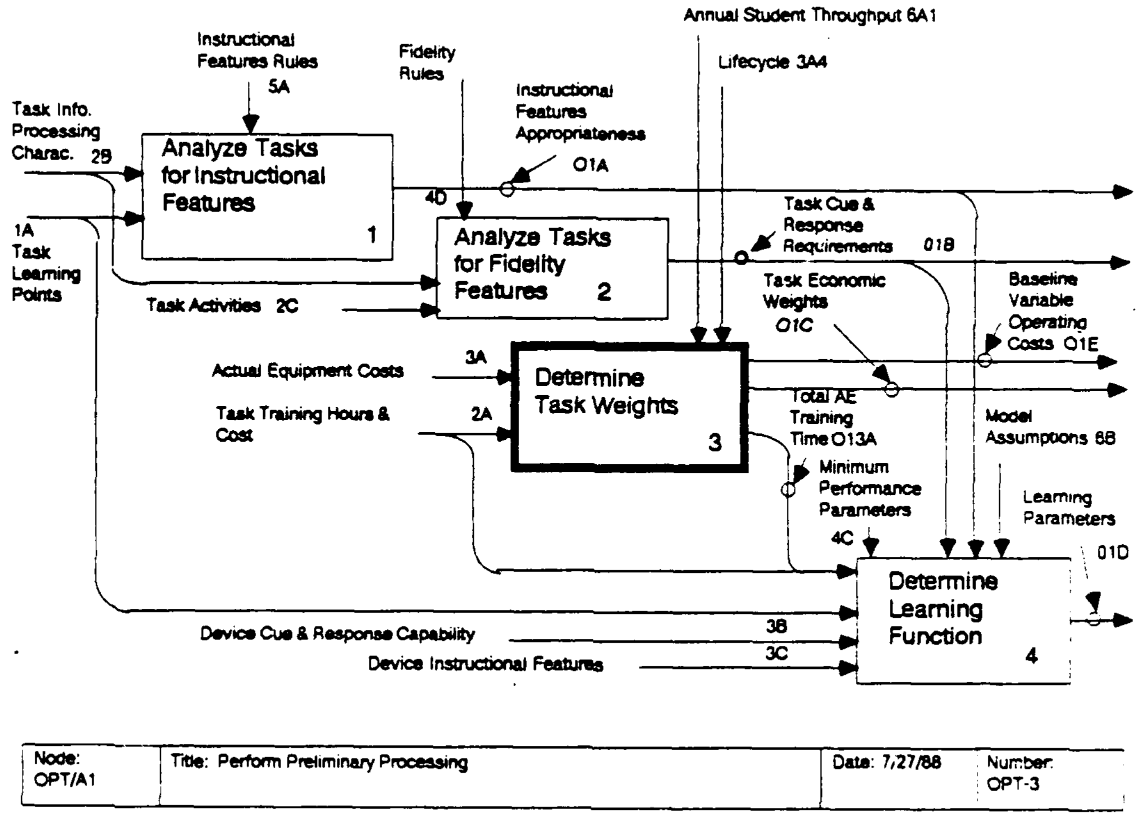
$$N_{AE} = \text{ceiling}\left(\left(\sum_T TAE_T\right) * GRADS/UMAX_{AE}\right),$$

where

ceiling(argument) = the smallest integer at least as great as the function argument.

The proportion of actual equipment hours per task ($TAENRM_T$) is:

$$TAENRM_T = TAE_T / \left(\sum_T TAE_T\right)$$



The allocated annual investment cost to each task ($CINV_T$) is:

$CINV_T = (INV_{AE}/LC_{AE}) * N_{AE} * TAENRM_T.$

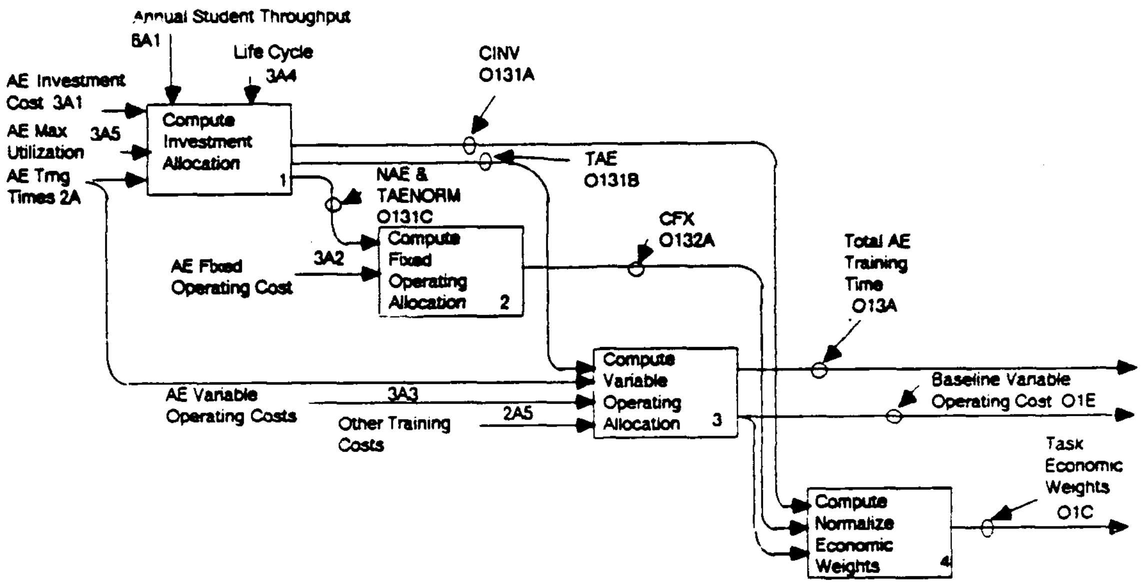
<u>OPT/A132: Compute the fixed operating cost allocation</u>. The allocation of fixed operating cost of the actual equipment (CFX_T) is determined by proportioning the fixed operating cost (FOC_{AE}; 3A2) over the tasks using the device in accordance to their usage. (Classroom fixed operating cost is assumed negligible when apportioned on a per-student basis.) That is,

$CFX_T = FOC_{AE} * N_{AE} * TAENRM_T.$

<u>OPT/A133: Compute the variable operating cost allocation</u>. The yearly variable operating cost, $CVAR_T$, allocated to training task T is just the total variable operating cost for each task, including other task training costs (such as ammunition):

$CVAR_T = [(VHRC_{T,AE} * TTOT_T) + OTC_T] * GRADS$

where $TTOT_T$ is the total number of training hours on the actual equipment (including the classroom) in order to train task T from entry to standard without use of a simulator, $VHRC_{T,AE}$ is the variable cost (per hour) of both the classroom and the actual equipment for task T, and OTC_T represents other costs of training



Node: OPT/A13	Title: Determine Task Weights	Date: 7/27/88	Number: OPT-4

task T, such as ammunition. OTC_T is available from the data base (2A5). $VHRC_{T,AE}$ and $TTOT_T$ are computed as follow:

$$TTOT_T = CRH_T + TAE_T$$

$$VHRC_{T,AE} = [(CRH_T * VHRC_{CR}) + (NOH_T * VHRC_{AE(NO)}) + ((OH_T + SUH_T) * VHRC_{AE(O)})]/TTOT_T$$

where the following are found in the data base or estimated using ad hoc rules:

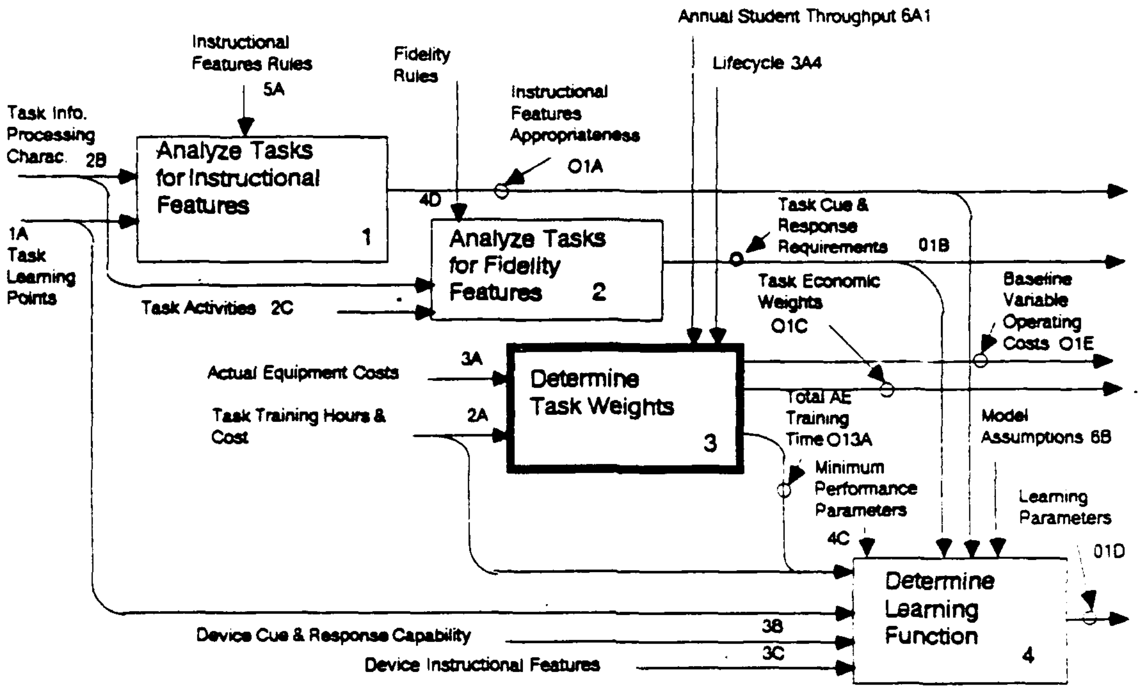
CRH_T = Number of classroom hours needed to train task T in conjunction with the actual equipment from entry to standard without a simulator (2A1)

$VHRC_{AE(NO)}$ = Variable cost (per hour) of the actual equipment in the non-operational training mode (assumed for now to be six times $VHRC_{CR}$)

$VHRC_{AE(O)}$ = Variable cost (per hour) of the actual equipment in the operational training mode (3A3)

$VHRC_{CR}$ = Variable cost (per hour) of the classroom (3A3).

<u>OPT/A134: Compute and normalize the economic weights</u>. The investment, fixed operating and variable operating allocations are summed to yield the total life cycle training cost for each

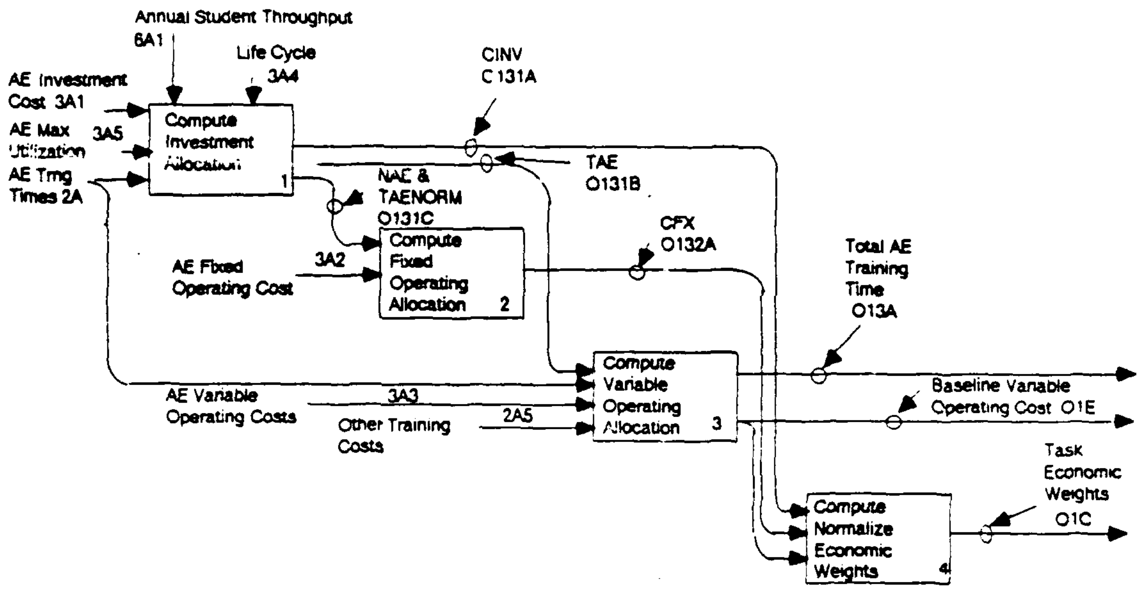


Node: OPT/A1	Title: Perform Preliminary Processing	Date: 7/27/88	Number: OPT-3

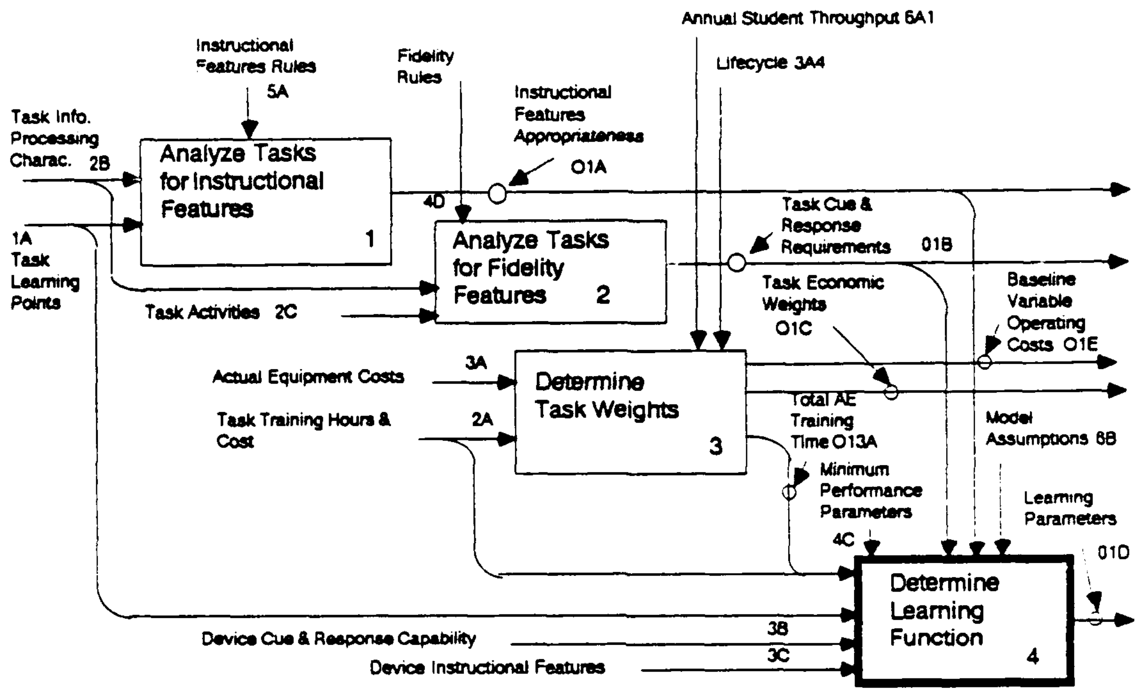
task ($TCST_T$), which is then normalized to sum to one across all tasks to produce the task weights (TWT_T).

$$TCST_T = CINV_T + CFX_T + CVAR_T$$

$$TWT_T = TCST_T / \left(\sum_T TCST_T \right)$$



Node: OPT/A13	Title: Determine Task Weights	Date: 7/27/88	Number: OPT-4



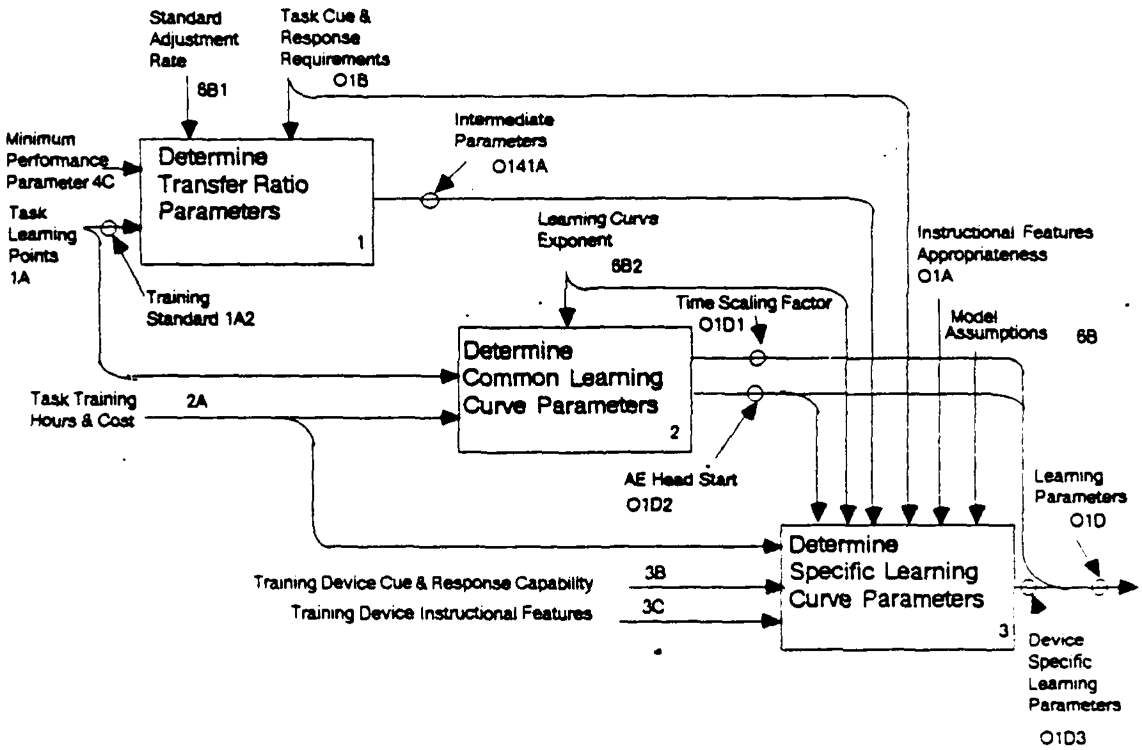
Node: OPT/A1	Title. Perform Preliminary Processing	Date: 7/27/88	Number: OPT-3

OPT/A14: Determine Learning Function

The parameters of the learning function depend upon the entry performance level and training standard (1A), the estimated training time on actual equipment (2A), task cue and response requirements (O1B), training device cue and response capabilities (3B) and instructional features (3C), the instructional feature appropriateness matrix (O1A), and several assumptions (6B). The learning curve parameters are estimated in three steps. In the first step, several intermediate values that are used to calculate transfer of training are determined. In the second step, the time scaling factor and actual equipment head start are determined. The results of the first two steps are used in the third step to determine device learning curves.

OPT/A141: Determine transfer ratio parameters. The asymptotes of the device learning curves, ASM_{Ti}, will be calculated in a later activity (OPT/A1431) according to the following equation:

$$ASM_{Ti} = q_T \left[\prod_j (FCAP_{ij} + b_{Tj})^{FRQT_{Tj}} \right]^{Q_T}$$



Node: OPT/A14	Title: Determine Learning Function	Date: 8/1/88	Number: OPT-5

where the Greek letter "pi" indicates a product (multiplication) of all the terms that follow, and where

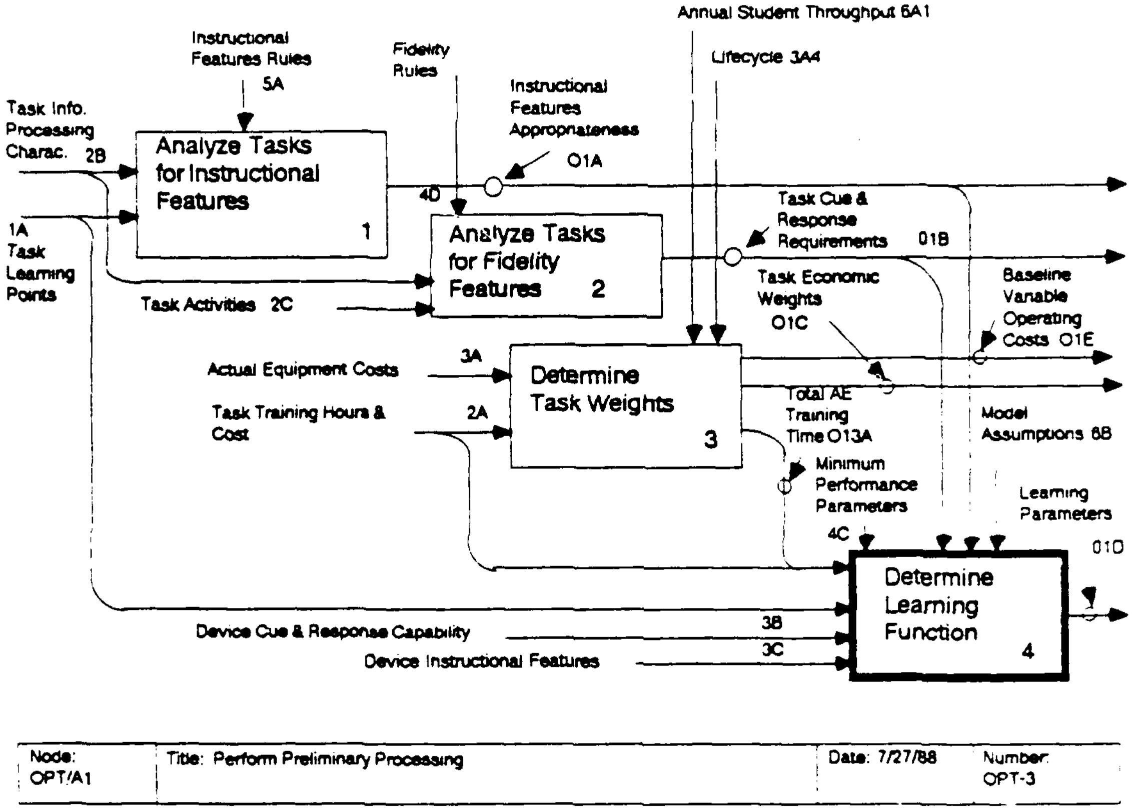
$FCAP_{ij}$ = the cue and response capability of device i on fidelity dimension j,

$FRQT_{Tj}$ = the cue and response requirement of task T on fidelity dimension j, and

q_T, b_{Tj}, and Q_t = scaling constants to adjust the range of the result.

The scaling constants adjust the result to be in the range [0,1], such that: (1) Perfect fidelity in all dimensions leads to an asymptote of 1.0; (2) Meeting all requirements of a task exactly produces an asymptote equal to the training standard adjusted by a certain percentage (this benchmark training standard is calculated in OPT/A1411); (3) A device with perfect fidelity in all dimensions except one, and no capability in the remaining dimension will produce an asymptote equal to the minimum performance parameter for that dimension (4C).

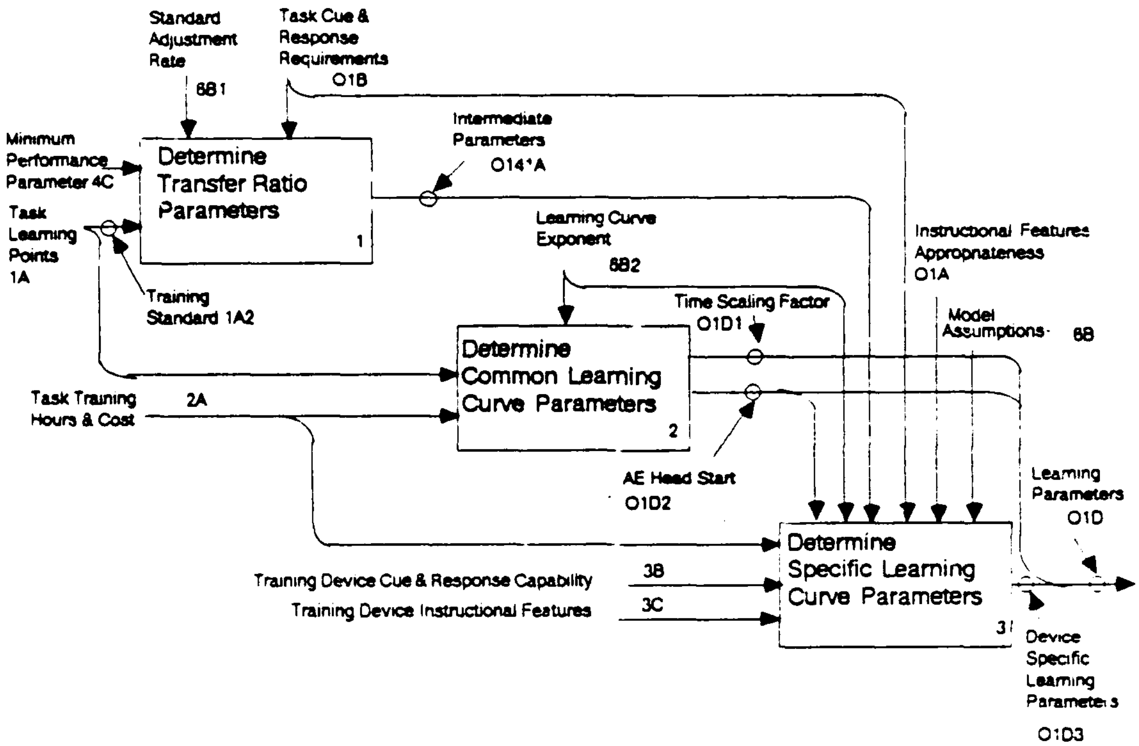
This activity calculates the scaling constants that will be used in the equation to calculate the learning curve asymptote. The specific procedures used in this activity estimate the benchmark transfer (OPT/A1411), equation exponent (OPT/A1412), reference point (OPT/A1413), and equation multiplier (OPT/A1414).



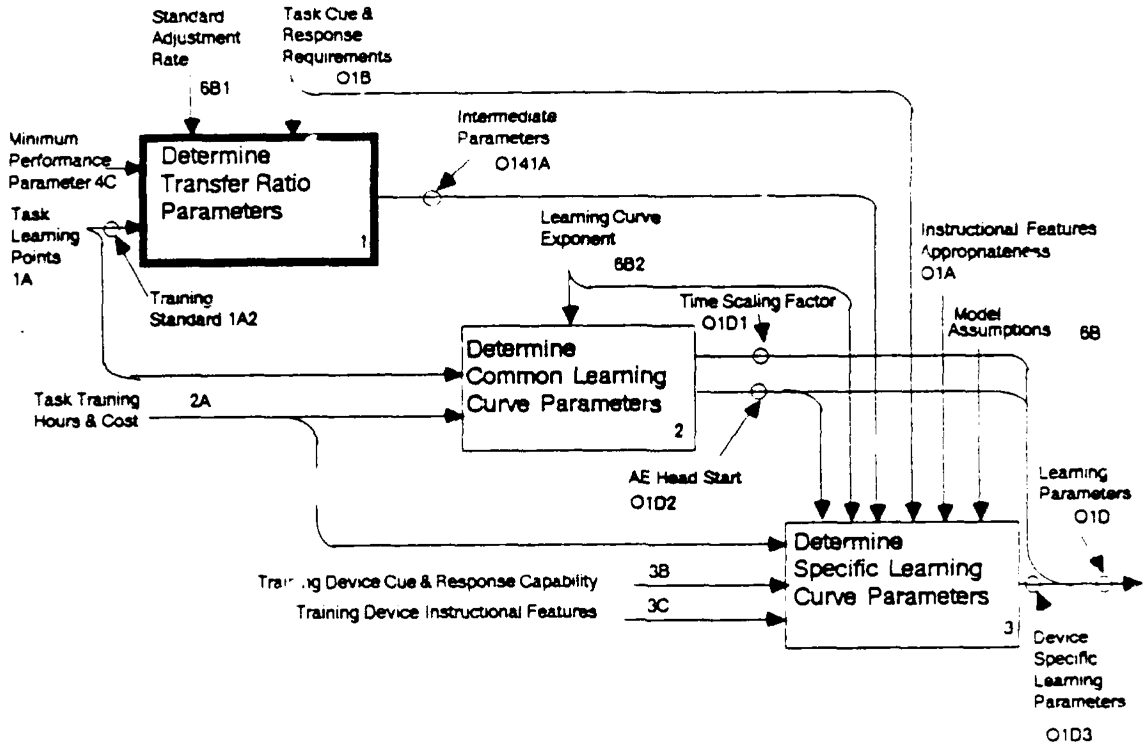
A mathematical derivation of the equations that are used is provided in Appendix B.

<u>OPT/A142: Determine common learning parameters</u>. This activity calculates the value of the time scaling factor, TSF_T, (in OPT/A1421) and the head start for actual equipment, $HS_{T,AE}$, (in OPT/A1422) to give an empirical fit to available training program data. The specific procedure is described in the detailed diagram for this activity.

<u>OPT/A143: Determine specific learning curve parameters</u>. This activity calculates the device-specific parameters of the learning curve. First, it applies the equation described in OPT/A14 to calculate the asymptote of the device curves (OPT/A1431). Then it calculates the time multiplier (TM_{Ti}) and head start (HS_{Ti}) for each training device (OPT/A1432 and OPT/A1433, respectively). The specific procedure is described in the detailed diagram for this activity.



Node: OPT/A14	Title: Determine Learning Function	Date: 8/1/88	Number: OPT-5

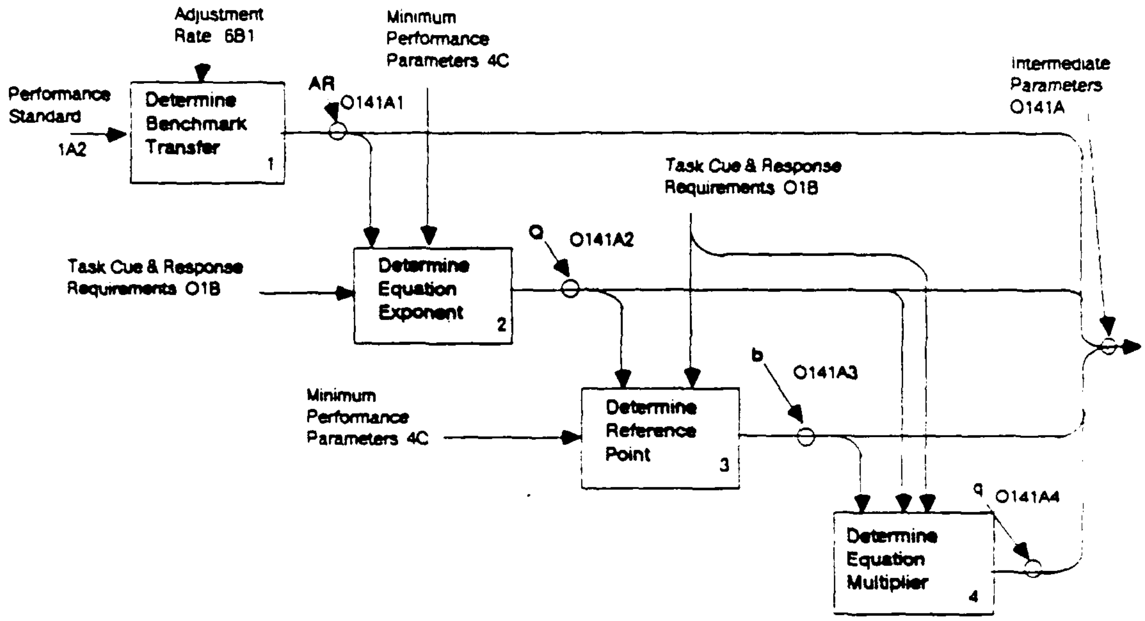


Node: OPT/A14	Title: Determine Learning Function	Date: 8/1/88	Number: OPT-5

OPT/A141: Determine Transfer Ratio Parameters

The parameters of the equation used to determine the asymptote of the device learning curves are determined in four steps. First a benchmark transfer value (ASM_{TR}) is determined by adjusting the training standard. Then the exponent of the equation (Q_T) is determined numerically to ensure that the asymptote for a device that exactly meets all cue and response requirements equals ASM_{TR}. Then the fidelity dimension reference points (b_{Tj}) are determined to set the maximum effect each fidelity dimension may have on the ASM_{Ti}. Finally, the equation multiplier (q_T) is determined to normalize the equation to have the maximum value of 1.0.

The equation for ASM_{Ti} has several properties that make it desirable. (1) A training device that meets all cue and response requirements will produce an asymptote that allows it to be used to train to the training standard in finite time. (2) Fidelity dimensions with high requirements are more important in the determination of the asymptote than dimensions with low requirements. (3) The overall importance of a dimension may be limited by the values of b_{Tj}.



Node: OPT/A141	Title: Determine Transfer Ratio Parameters	Date: 7/27/88	Number: OPT-8

<u>OPT/A1411: Determine benchmark transfer</u>. The benchmark transfer (ASM_{TR}) is the asymptote that is associated with the hypothetical training device that exactly meets all cue and response requirements. It is obtained by adjusting the training standard by a constant proportion, as follows:

$$ASM_{TR} = STD_T + SADJ\ (1 - STD_T)$$

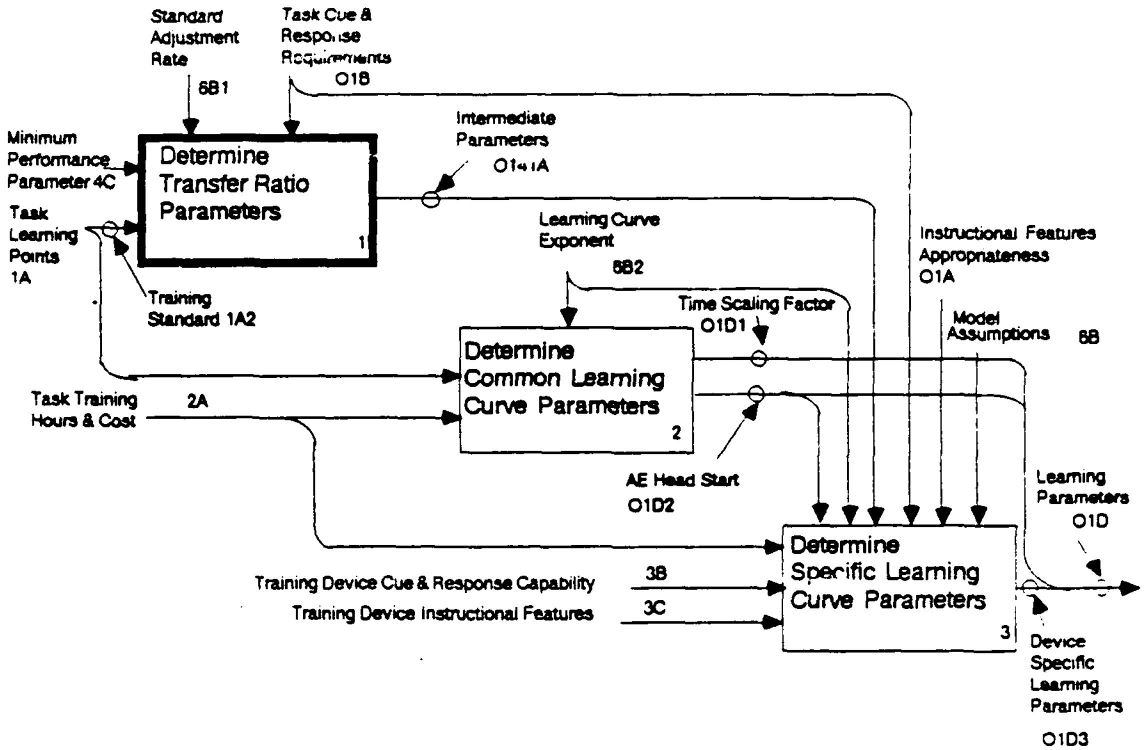
where

STD_T = the training standard for task T, and
SADJ = the adjustment proportion.

<u>OPT/A1412: Determine equation exponent</u>. The exponent of the equation (Q_T) is set to ensure that the asymptote for a device that exactly meets all cue and response requirements is ASM_{TR}. This condition may be expressed algebraically as follows:

$$ASM_{TR} = \prod_{j} \left[FRQT_{Tj} + (1 - FRQT_{Tj})\ FMIN_j^{\frac{1}{FRQT_{Tj}\ Q_T}} \right]^{FRQT_{Tj}\ Q_T}$$

where



Node: OPT/A14	Title: Determine Learning Function	Date: 8/1/88	Number: OPT-5

$FMIN_j$ = the minimum performance of a device with cue and response capability of 0.0 on dimension j, 1.0 on all other dimensions, and

$FRQT_{Tj}$ = the cue and response requirement for task T on dimension j.

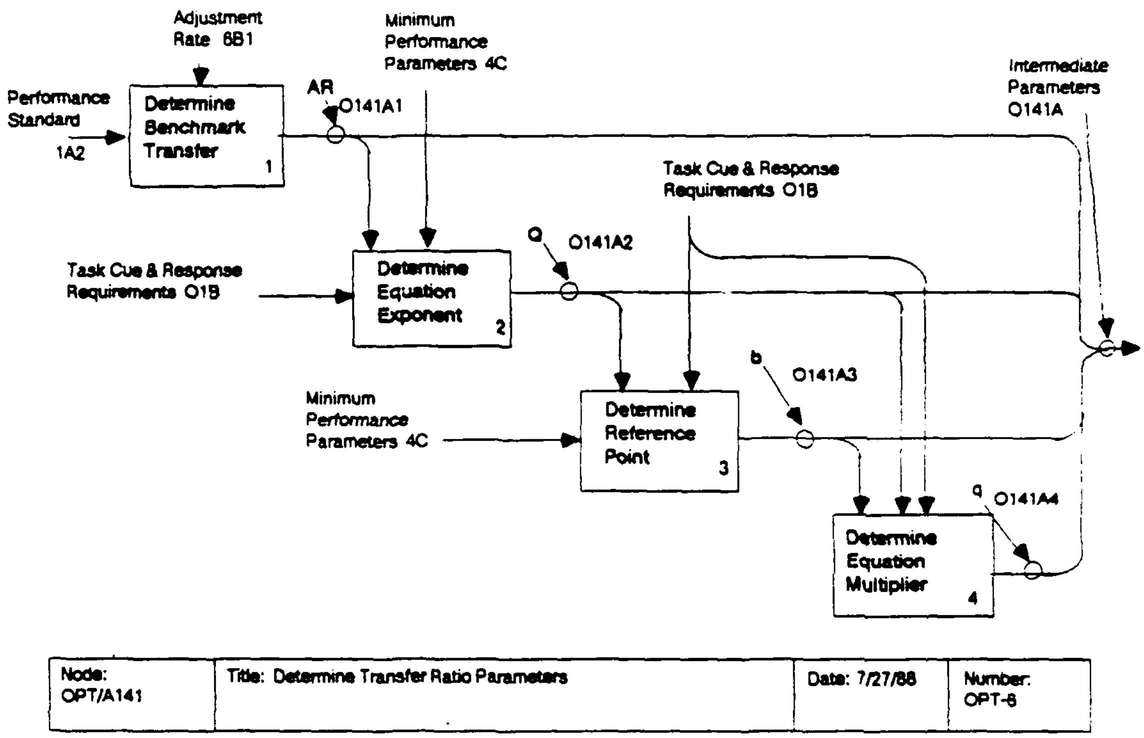
The solution may be found numerically using bisection. However, when

$$ASM_{TR} < \prod_j FMIN_j^{\,1 - FRQT_{Tj}}$$

then the $FMIN_j$ and the $FRQT_{Tj}$ are inconsistent, and no value of Q_T exists. In this case, the $FMIN_j$ must be lowered for tasks with inconsistencies until a solution is found. To eliminate the inconsistencies, the $FMIN_j$ are multiplied by the following factor:

$$0.8^{1/JC_T} * FMIN_j \div \left[\left[\frac{\prod_j FMIN_j^{(1 - FRQT_{Tj})}}{ASM_{TR}} \right]^{1/JC_T} \right]^{1/(1-FRQT_{Tj})}$$

where



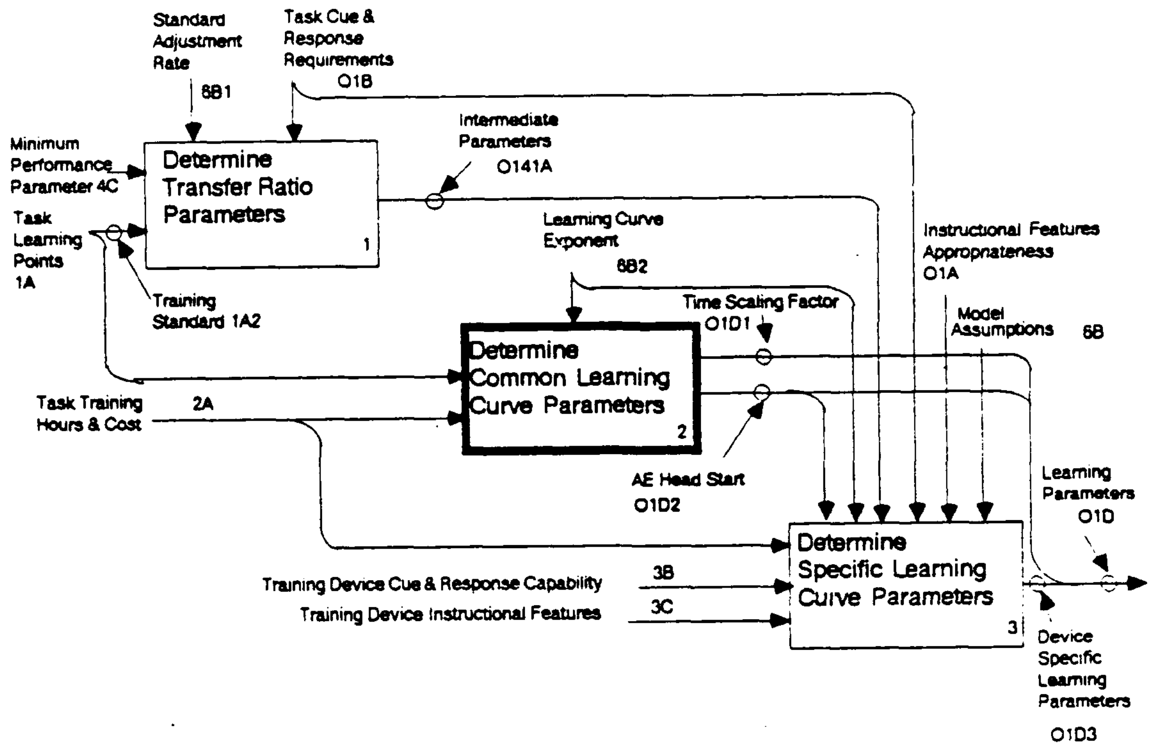
JC_T = the number of fidelity dimensions, j, such that $FRQT_{Tj} > 0$ and $FMIN_j > 0$.

OPT/A1413: Determine reference point. The cue and response reference point (b_{Tj}) is a variable that is included in the asymptote calculation so that the asymptote is $FMIN_j$ in the appropriate situations. The b_{Tj} are calculated using the following formula:

$$b_{Tj} = \frac{FMIN_j^{\frac{1}{FRQT_{Tj}\ Q_T}}}{1 - FMIN_j^{\frac{1}{FRQT_{Tj}\ Q_T}}}$$

OPT/A1414: Determine equation multiplier. The multiplier of the equation that determines the asymptote, q_T, normalizes the equation to have a maximum value of 1.0. The multiplier is calculated using the following equation:

$$q_T = \left[\prod_j (1 + b_{Tj})^{FRQT_{Tj}} \right]^{-Q_T}$$



Node: OPT/A14	Title: Determine Learning Function	Date: 8/1/88	Number: OPT-5

OPT/A142: Determine Common Learning Curve Parameters

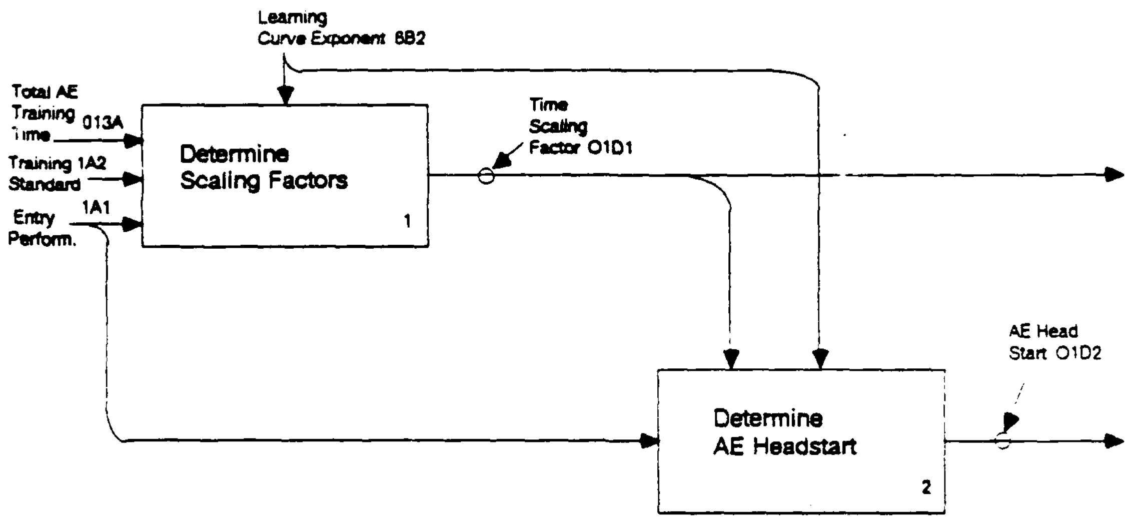
This activity calculates the time scaling factor (TSF_T) for each task and the head start (HS_{Ti}) for actual equipment.

OPT/A1421: Determine time scaling factor. By convention, the time multiplier parameter, TM_i for actual equipment is 1. Training data for actual equipment are used to calculate the time scaling factor, TSF_T, using the following formula. We assume that the learning curve exponent, r, is 0.7 to be consistent with the human factors literature (e.g., Card, Moran, & Newell, 1983) and with theoretical considerations (Newell & Rosenbloom, 1981).

$$TSF_T = \frac{(1 - STD_T)^{-1/r} - (1 - ENT_T)^{-1/r}}{TTOT_T}$$

where

ENT_T = the entry level performance on task T
STD_T = the training standard for task T, and
$TTOT_T$ = the total number of training hours on the actual equipment to train to the standard on task T.



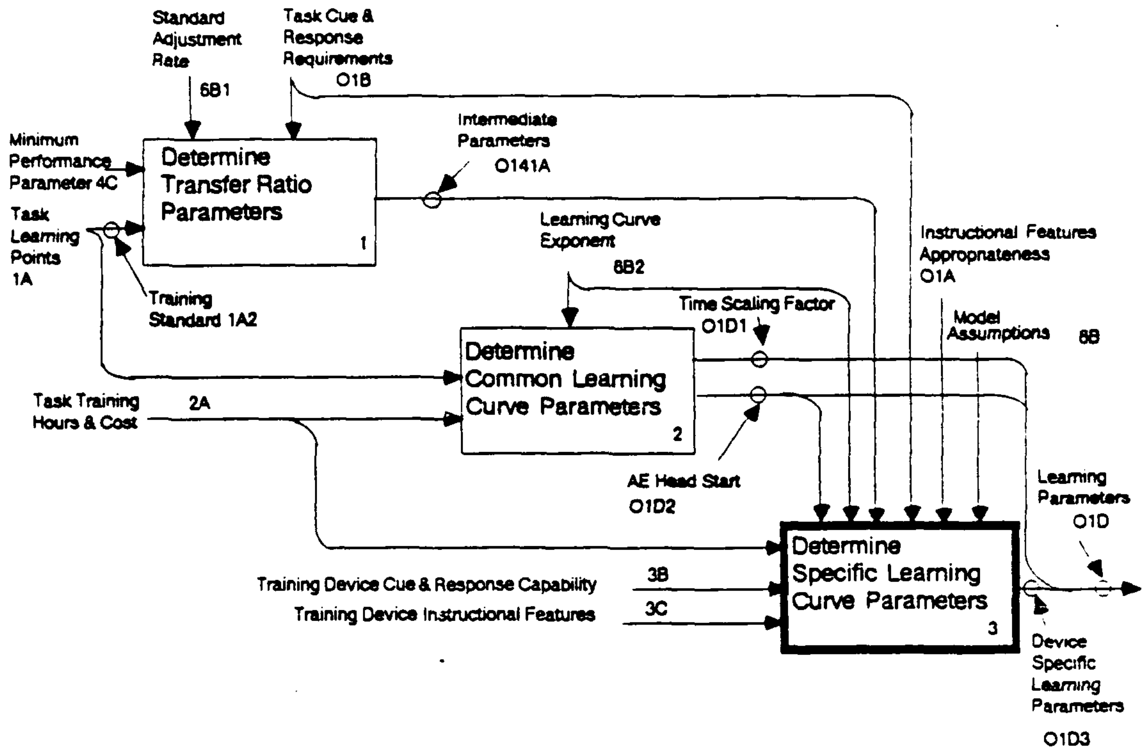
Node: OPT/A142	Title: Determine Common Learning Curve Parameters	Date: 7/28/88	Number: OPT-7

The time scaling factor is related to the movement in performance between entry and the training standard that is associated with an hour of training time on the actual equipment.

OPT/A1422: Determine actual equipment head start. The head start for actual equipment ($HS_{T,AE}$) may be calculated in this node because both TM_{Ti} and ASM_{Ti} are equal to 1.0 for the actual equipment. Under these conditions, the following formula is equivalent to the formula presented in OPT/A1433.

$$HS_{T,AE} = \frac{1}{TSF_T} \left[(1 - ENT_T)^{-1/r} - 1 \right]$$

With this head start included in the learning curve for actual equipment, a student who trains for zero time will still score performance ENT_T on task T.



Node: OPT/A14	Title: Determine Learning Function	Date: 8/1/88	Number: OPT-5

OPT/A143: Determine Specific Learning Curve Parameters

This activity calculates the learning curve parameters that are specific to both the task and the training device. These parameters include the asymptote (ASM_{Ti}), the time multiplier (TM_{Ti}), and the head start (HS_{Ti}). Each of the following three subactivities determines one of these parameters.

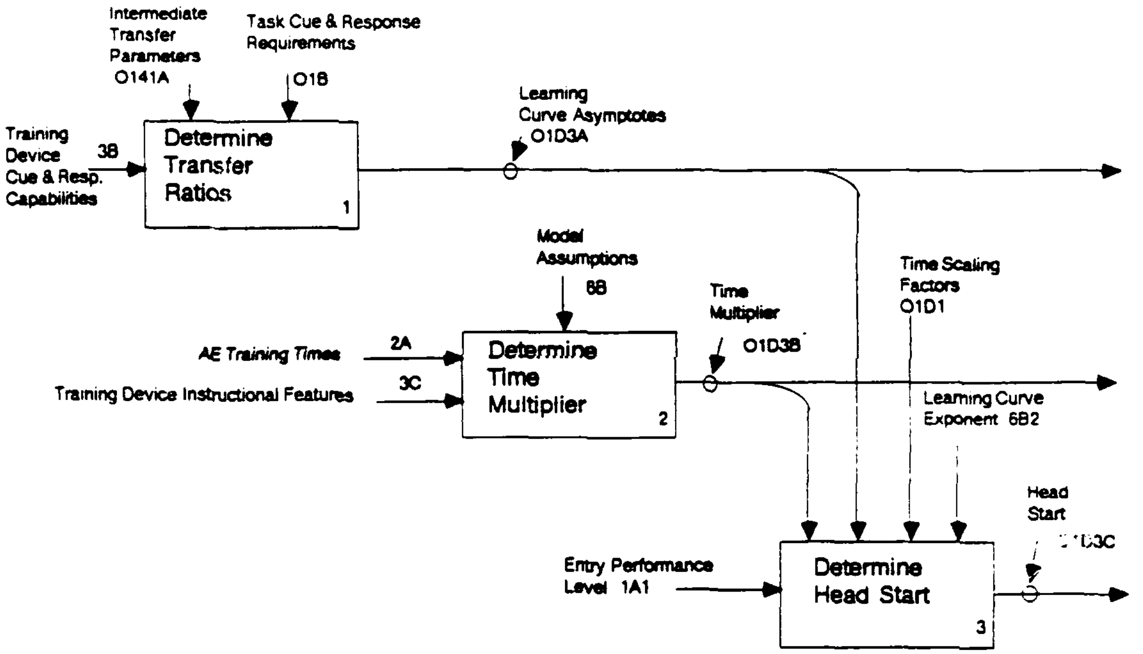
OPT/A1431: Determine transfer ratio. The asymptote or transfer ratio (ASM_{Ti})is calculated from the transfer parameters determined in OPT/A141, the task cue and response requirements ($FRQT_{Tj}$), and the device cue and response capabilities ($FCAP_{ij}$), according to the following formula:

$$ASM_{Ti} = q_T \left[\prod_j (FCAP_{ij} + b_{Tj})^{FRQT_{Tj}} \right]^{Q_T}$$

where

$FCAP_{ij}$ = the cue and response capability of device i on fidelity dimension j,

$FRQT_{Tj}$ = the cue and response requirement of task T on fidelity dimension j,



Node: OPT/A143	Title: Determine Specific Learning Curve Parameter	Date: 7/28/88	Number: OPT-8

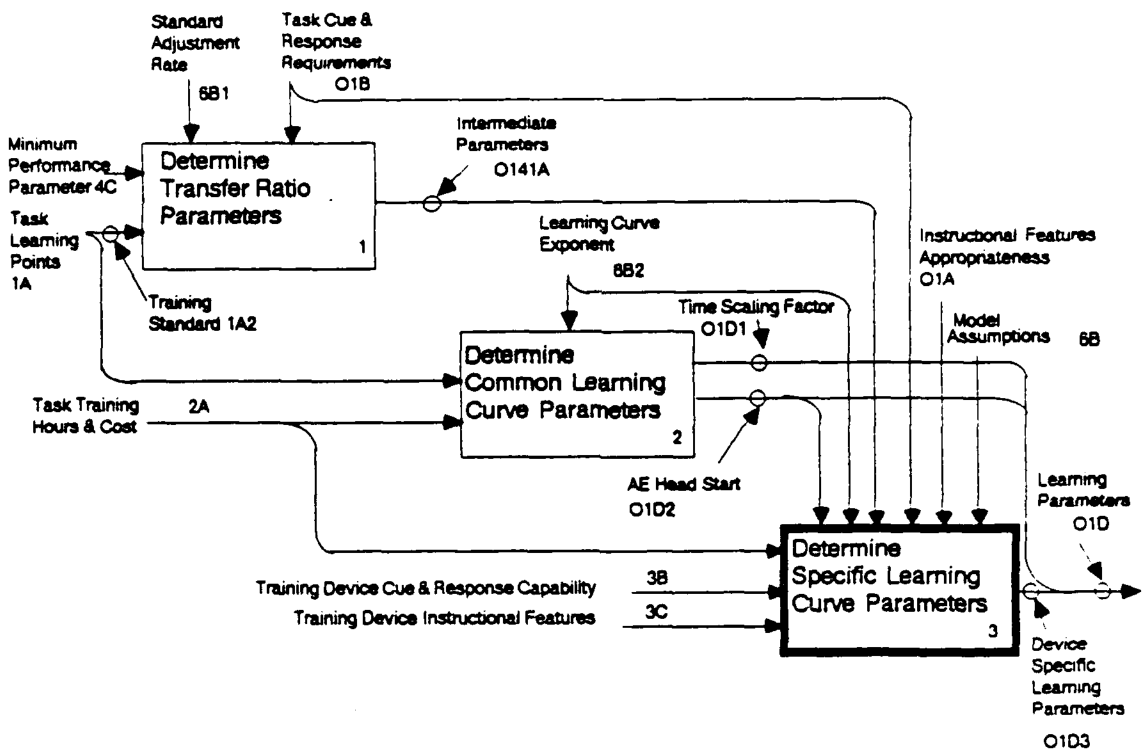
q_T = the equation multiplier (O141A4),
b_{Tj} = the fidelity dimension reference point (O141A3), and
Q_t = the equation exponent (O141A2).

<u>OPT/A1432: Determine time multiplier</u>. The time multiplier reflects the improvement in training efficiency that may occur through the use of a training device. This improvement may occur because setup time is reduced, or may result from the proper use of instructional features. Several assumptions are used to predict the extent of efficiency improvement. The processes in this activity combine the effect of instructional features (determined in OPT/A14321) and reduced setup time (determined in OPT/A14322 and OPT/A14323) to determine the time multiplier (in OPT/A14324). The detailed diagram for this activity describes both the procedure and assumptions used in this activity.

<u>OPT/A1433: Determine head start</u>. The head start (HS_{Ti}) is set to ensure that the entry performance level is obtained for the time, t = 0. HS_{Ti} is calculated using the following formula:

$$HS_{Ti} = \frac{1}{TM_{Ti}\ TSF_T}\left[(1 - ENT_T/ASM_{Ti})^{-1/r} - 1\right]$$

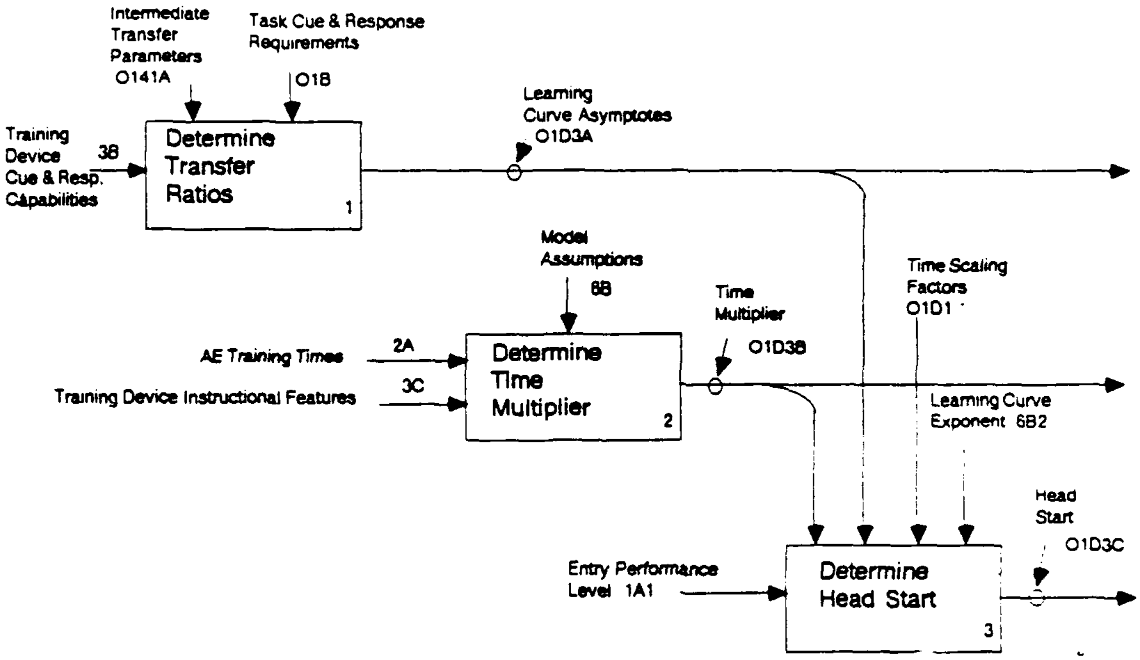
where $ENT_T < ASM_{Ti}$, and



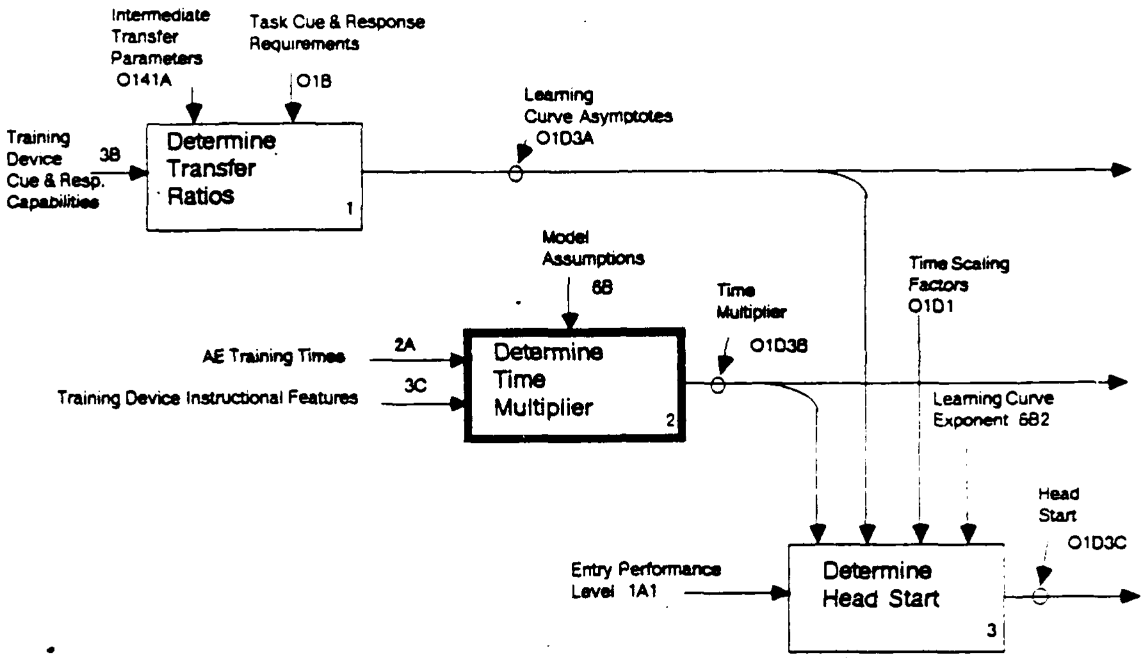
Node: OPT/A14	Title: Determine Learning Function	Date: 8/1/88	Number: OPT-5

TM_{Ti} = the time multiplier for task T and device i (O1D3B),
TSF_T = the time scaling factor for task T (O1D1),
ENT_T = the entry performance level for task T (1A1),
ASM_{Ti} = the asymptote for task T and device i (O1D3A), and
r = the exponent of the learning function (assumed to be 0.7).

HS_{Ti} is undefined when $ENT_T > ASM_{Ti}$, because this condition indicates that is impossible to obtain operational performance even as great as the entry performance level using training device i.



Node: OPT/A143	Title: Determine Specific Learning Curve Parameter	Date: 7/28/88	Number: OPT-8

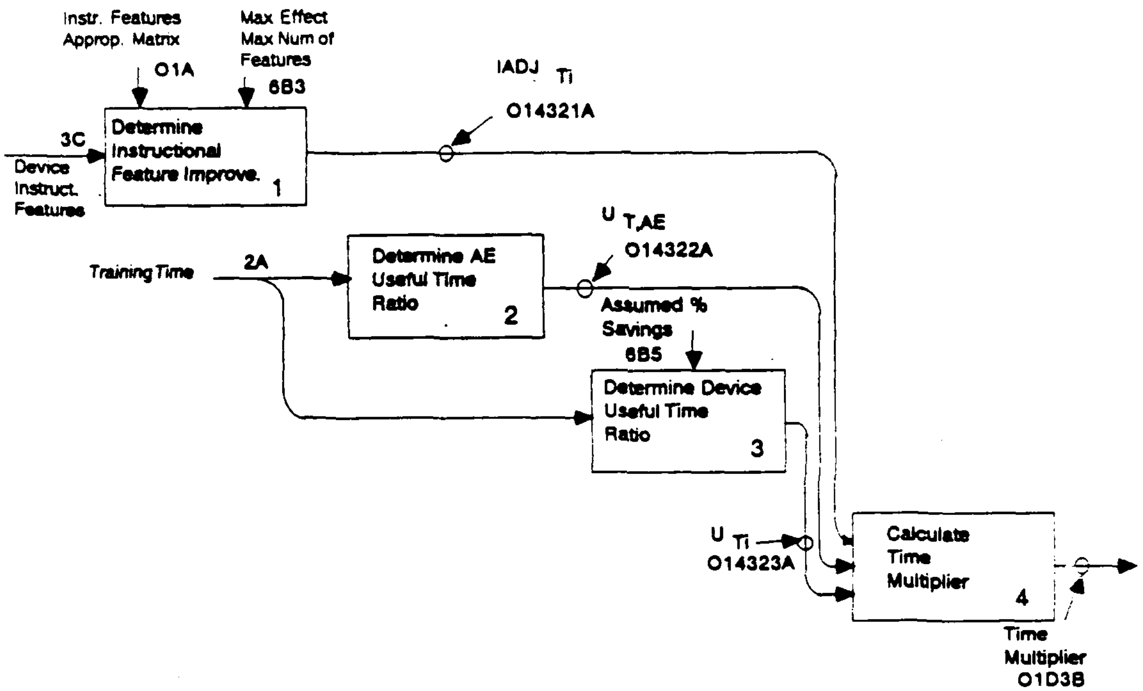


Node: OPT/A143	Title: Determine Specific Learning Curve Parameter	Date: 7/28/88	Number: OPT-8

OPT/A1432: Determine Time Multiplier

The time multiplier (TM_{Ti}) measures the increased efficiency of device-based training due to instructional features and reduced setup time. This activity is not carried out for actual equipment, since the time multiplier for actual equipment is 1.0.

OPT/A14321: Determine instructional feature improvement. The estimated instructional feature improvement in efficiency ($IADJ_{Ti}$) is affected by two limits. The first limit reflects the maximum number of appropriate instructional features that would be of benefit to training (MAXIF; 6B4). A device with more than MAXIF instructional features would not get any additional benefit over a device with MAXIF features. The second limit is the maximum percentage improvement in training efficiency due to instructional features (IFBN; 6B3). A device with MAXIF appropriate instructional features would produce a proportional decrease in training time of IFBN. The first formula in the calculation of $IADJ_{Ti}$ calculates the number of appropriate instructional features from the instructional feature appropriateness matrix, IFT_{Tk}, (O1A) and the list of instructional features possessed by each device, $DVIF_{ik}$ (3C).



Node: OPT/A1432	Title: Determine Time Multiplier	Date: 7/28/88	Number: OPT-9

$$IFNM_{Ti} = \min \left[\left(\sum_{k} IFT_{Tk}\ DVIF_{ik} \right), MAXIF \right]$$

The final adjustment value ($IADJ_{Ti}$) is then determined using the following formula.

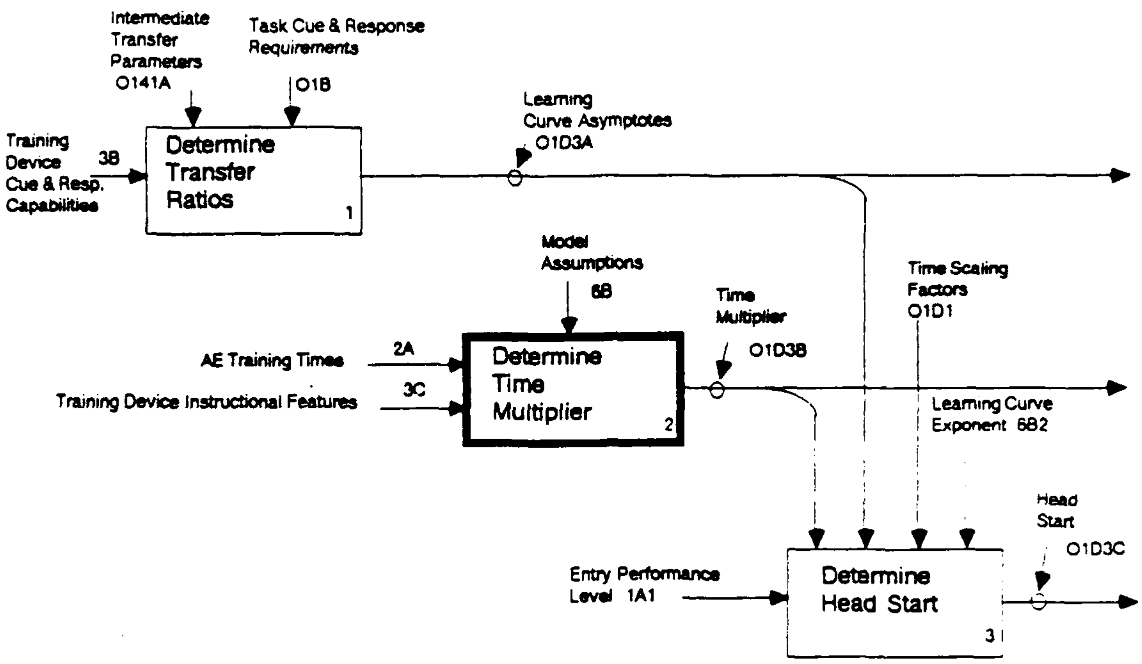
$$IADJ_{Ti} = \frac{1}{1 - \left[(IFNM_{Ti}/MAXIF)^{1/2}\ IFBN \right]}$$

This equation produces an adjustment such that each additional instructional feature has a smaller impact on $IADJ_{Ti}$ than the previous instructional feature.

<u>OPT/A14322: Determine actual equipment useful time ratio</u>. The actual equipment useful time ratio ($U_{T,AE}$) represents the percentage of use of the equipment in which effective training occurs. That is,

$$U_{T,AE} = \frac{CRH_T + NOH_T + OH_T}{TTOT_T} .$$

The numerator in this expression represents the time the student is actually undergoing instruction in the classroom (CRH_T), on



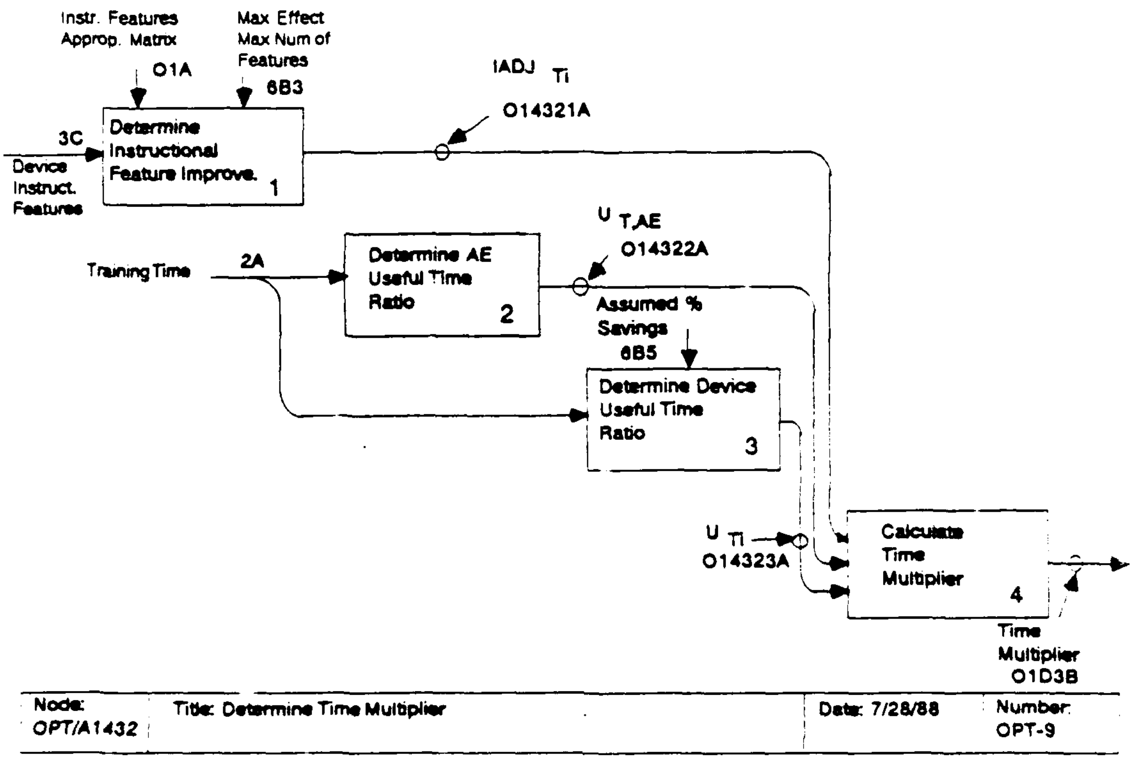
Node: OPT/A143	Title: Determine Specific Learning Curve Parameter	Date: 7/28/88	Number: OPT-8

the actual equipment prior to operation (NOH_T), and on the actual equipment during operation (OH_T). The denominator represents this total instruction time plus setup hours, SUH_T. Setup hours include such things as student time in being transported from classroom to airfield, checking out flight equipment, waiting for takeoff clearance, or performing tasks that are already known to the performance standard, i.e., obligatory student time during which no instruction is being provided.

<u>OPT/A14323: Determine device useful time ratio</u>. The device useful time ratio (U_{Ti}) represents the percentage of use of the device in which effective training occurs. The formula used assumes that some percentage (SUSAV; 6B5) of the setup time is saved when training is conducted in a training device. That is,

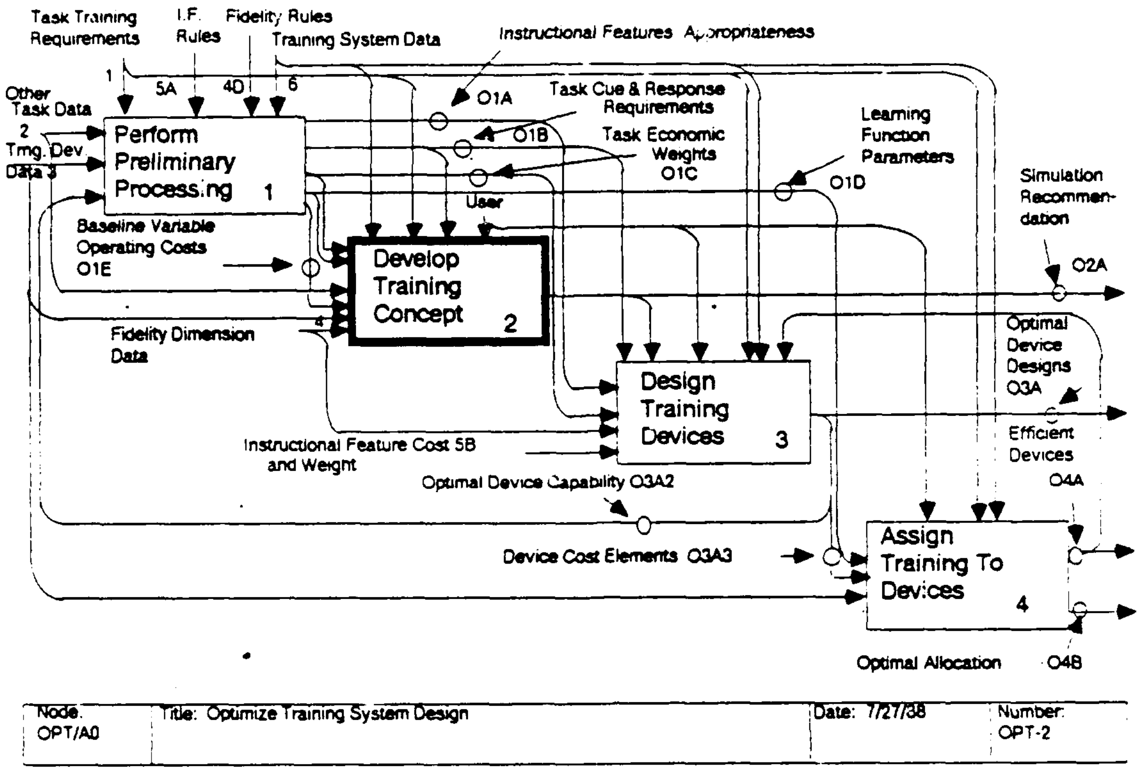
$$U_{Ti} = \frac{CRH_T + NOH_T + OH_T}{TTOT_T - SUH_T * SUSAV}$$

<u>OPT/A14324: Calculate time multiplier</u>. The time multiplier (TM_{Ti}) is calculated from the previous results using the following formula:



$$TM_{Ti} = IADJ_{Ti} \; \frac{U_{Ti}}{U_{T,AE}} .$$

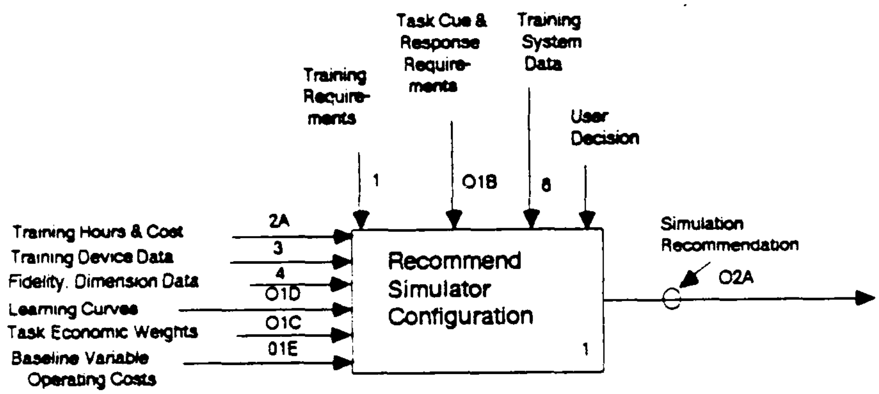
It represents an improvement in the usage of time, compared to the usage of time on actual equipment, due to improved instructional features ($IADJ_{Ti}$) or due to a reduction in "wasted" setup time ($U_{Ti}/U_{T,AE}$).



OPT/A2: Develop Training Concept

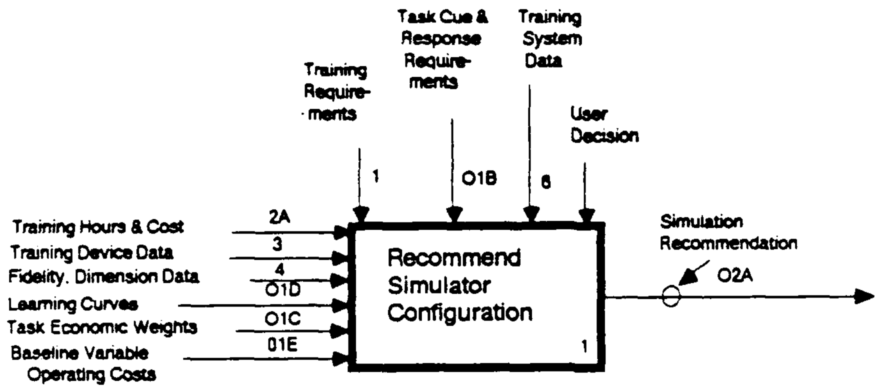
This component is concerned with methods to cluster tasks that have similar training-device needs. The output of this component is a set of preliminary training-device requirements that can be used as the basis of the training-device design process in OPT/A3. We have currently developed a single tool for this activity, a tool that evaluates general training system alternatives including full-mission simulators, part-mission simulators, and actual equipment. Because there is only one tool in this component, this level of decomposition in the IDEFO model is not required. However, it was included for two reasons. First, adding this level allows us to represent all tools at the same level of decomposition in the overall model. It is hoped that this parallel structure will make the numbering of model processes easier to understand. Second, including this level of decomposition provides a place holder for future tools that might be developed for this training-concept development.

OPT/A21: Recommend simulator configuration. This tool evaluates general training system alternatives. It considers the following three classes of training device: (1) full-mission simulators, (2) part-mission simulators, and (3) actual equipment. The tool evaluates these alternatives to provide guidance regarding the types of device that would be appropriate



Node: OPT/A2	Title: Develop Training Concept	Date: 7/12/88	Number: OPT-10

for the training requirements. The recommendations are based on the task requirements and estimates of the cost to meet these requirements. The activity contains two subactivities, evaluate each task (OPT/A211) and develop simulation recommendations (OPT/A212).

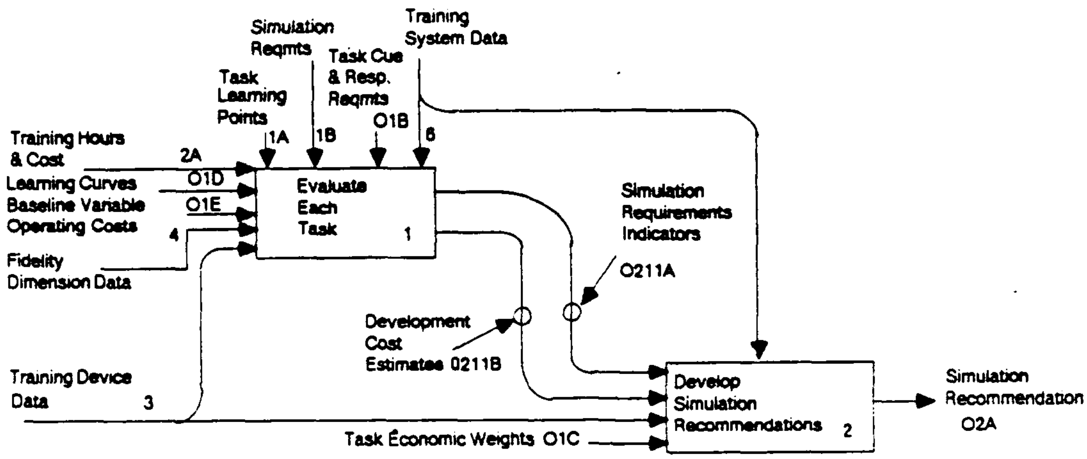


Node: OPT/A2	Title: Develop Training Concept	Date: 7/12/88	Number: OPT-10

OPT/A21: Recommend Simulator Configuration

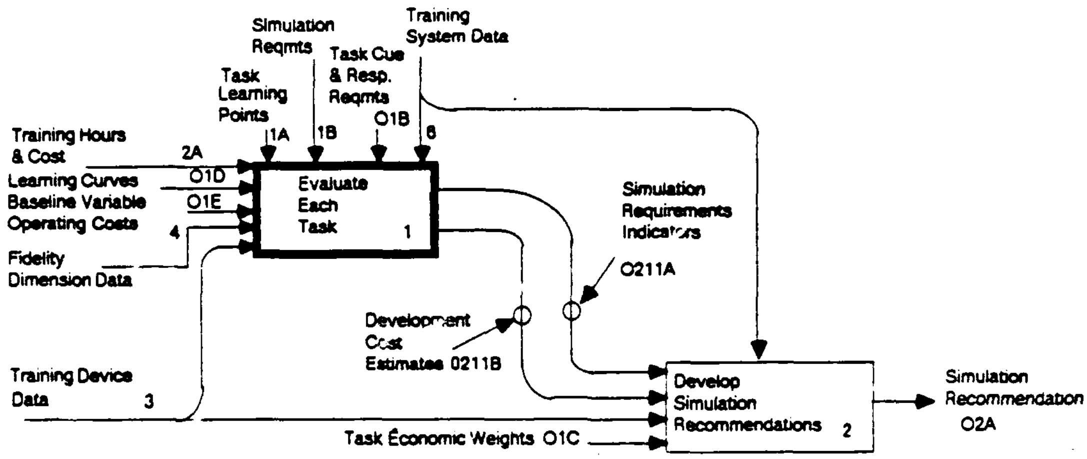
This tool examines the need and cost-effectiveness of using either a full-mission simulator (FMS) or one or more part-mission simulators (PMSs) to replace training time in the actual equipment. Built into the evaluation performed by this tool is the examination of simulator-unique capabilities, such as training in unsafe situations, and cost savings to establish the value of training with some sort of training device. In addition, the development cost of the training device that would be required to achieve these benefits is examined. Using these results the module partitions the set of training requirements into the following three subsets: (1) a subset for which an FMS should be designed by the later modules, (2) a subset requiring one or more PMSs to meet the training needs, (3) a subset requiring training on actual equipment.

OPT/A211: Evaluate each task. This first process is a set of operations that acts upon each task; it is repeated for every task. The requirements for training by simulator and the operating cost savings associated with simulator training (O211A) are first examined for each task (in OPT/A2111). Then the task cue and response requirements to achieve these results are used to provide a preliminary estimate of training-device development costs (O211B; evaluated in OPT/A2112).



Node: OPT/A21	Title: Recommend Simulator Configuration	Date: 7/12/88	Number: OPT-11

<u>OPT/A212: Develop simulation recommendations</u>. This second process then operates on the simulation requirement index (O211A) and development cost index (O211B) from the previous process to generate a recommendation concerning the need for a full-mission simulator versus several possible part-mission simulators. This recommendation compares the development cost index with the potential operating cost savings over the equipment lifecycle to recommend simulation-based or actual-equipment-based training. The development cost index is used to recommend training on an FMS or PMS for those tasks for which simulation-based training is indicated. The first subprocess of this activity (OPT/A2121) creates a scatterplot that shows the simulation requirement index and development cost index for each task. The second subprocess (OPT/A2122) summarizes the recommendations.



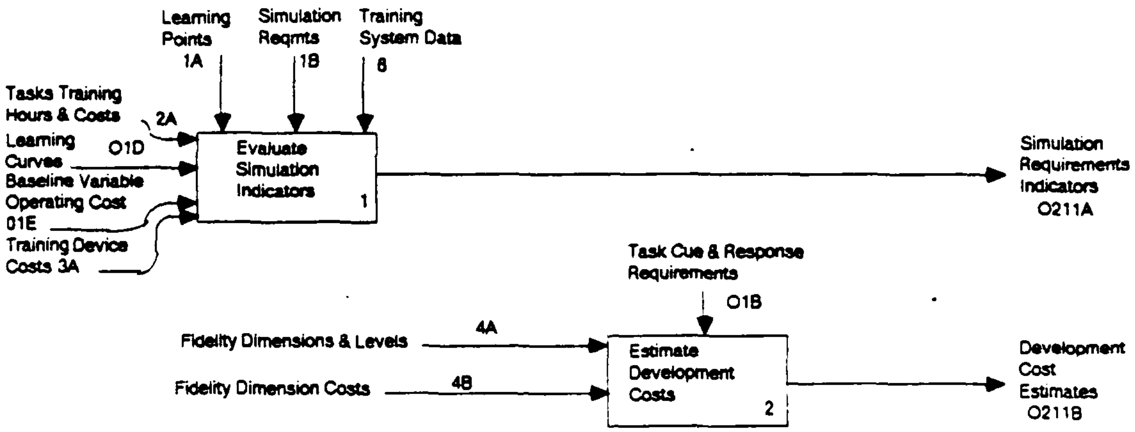
Node: OPT/A21	Title: Recommend Simulator Configuration	Date: 7/12/88	Number: OPT-11

OPT/A211: Evaluate Each Task

Evaluate simulation indicators and evaluate fidelity requirements are the two processes needed for this activity. These processes generate the data that are needed to develop simulation recommendations (OPT/A212).

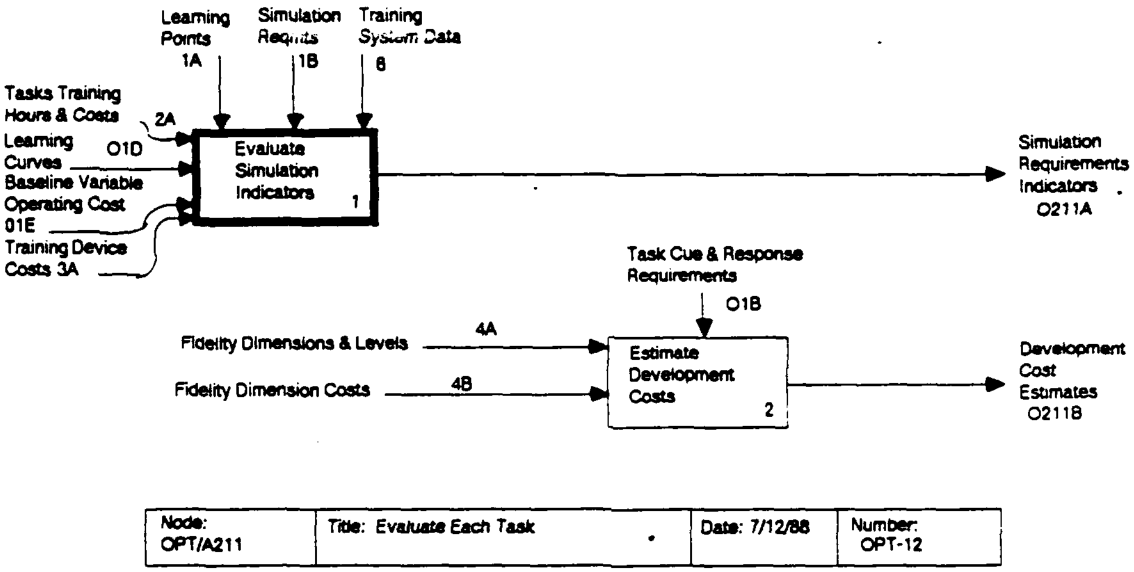
OPT/A2111: Evaluate simulation indicators. The two classes of simulation indicators, requirements and operating-cost savings, determine the extent to which a simulator of any kind would be beneficial for training on a task-by-task basis. These indices are developed by aggregating SME judgments on a variety of factors associated with simulator value. The indicators are used to justify expenses in the development of simulation capabilities. The simulation requirements are determined in OPT/A21111; the operating-cost savings is determined in OPT/A21112.

OPT/A2112: Estimate Development Costs. The fidelity requirements of each task quantify the relative need for a full-mission simulator compared to part-mission simulators. In addition, tasks that require a simulator to represent a variety of different cues and responses to a high level of fidelity will require a more expensive simulator to train effectively. Thus, the fidelity dimension data can be used to compute the



Node: OPT/A211	Title: Evaluate Each Task	Date: 7/12/88	Number: OPT-12

development cost required to train the task on a simulator. The activity proceeds by first estimating the cost to meet the requirements for each fidelity dimension (OPT/A21121) and then summing the costs across fidelity dimensions (OPT/A21122).



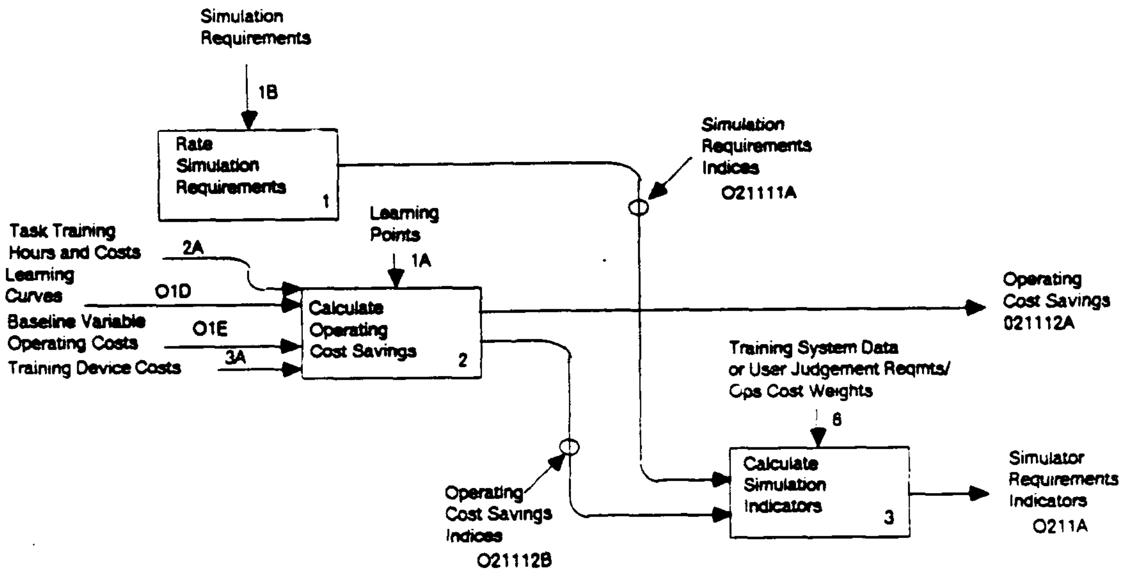
OPT/A2111: Evaluate Simulation Indicators

This analysis results in the quantification of a simulation indicator (O211A) for each task. This simulation indicator reflects both the requirement for simulation to train the task (O21111A) and the cost savings that would be expected if simulation were employed (O21112A).

OPT/A21111: Rate simulation requirements. This analysis is completed by using the data for simulation requirements (1B). These data were developed using the following 11 conditions:

(a) training without simulation is impossible

- task involves imposing an emergency (e.g., malfunction) that places the student and equipment in danger
- task involves a procedure that violates safety standards for the actual equipment
- task involves a violation of current regulations or restrictions on operations
- task involves response to specific battle situations that cannot be provided in the real world
- trainees inexperience makes task unsafe
- task involves response to a component failure that cannot be induced without undue hazard



Node: OPT/A2111	Title: Evaluate Simulation Indicators	Date: 7/12/88	Number: OPT-13

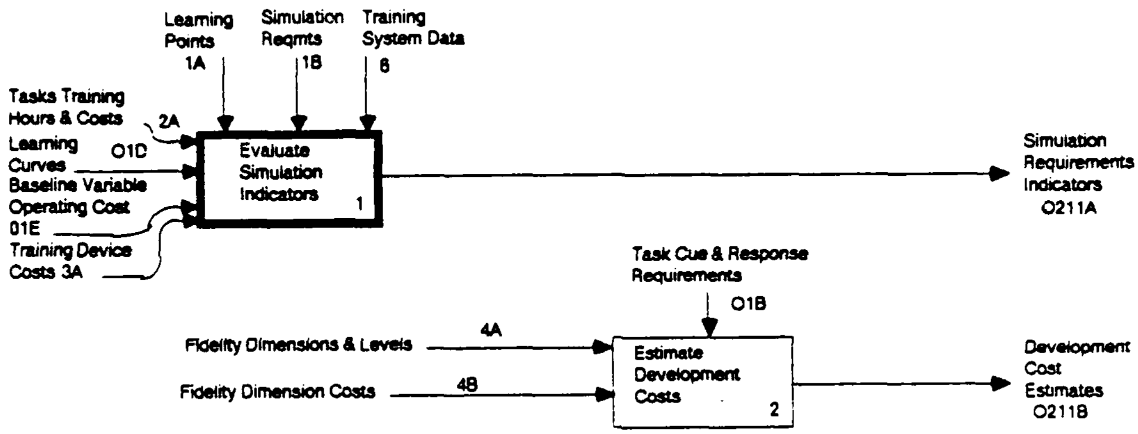
(b) training without simulation is difficult or costly

- task involves operation in special weather or lighting conditions that do not occur naturally often enough
- task involves response to specific battle situations or other special situations that are very expensive in the real world
- task involves training in equipment that is not accessible enough

(c) training would be significantly improved with simulation

- simulation offers special training effects that enhance acquisition or transfer
- number of trials per unit time can be greatly increased with simulation.

If any condition in group (a) pertains for a given task, simulation is considered an absolute requirement and a rating of 1.0 is assigned to the index for simulation requirements, $ISIM_T$. If no condition from group (a) pertains, the number of conditions in groups (b) and (c) determine the rating. One condition in group (b) yields a rating of 0.4, two yield a 0.6, and three yield a 0.7. One condition in group (c) is worth 0.2, and two

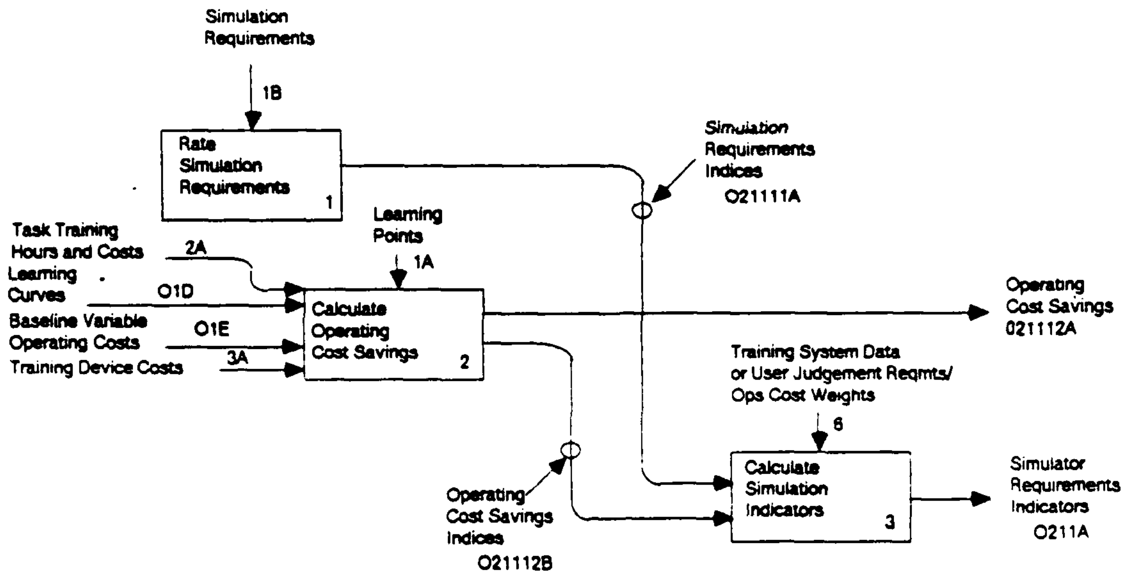


Node: OPT/A211	Title: Evaluate Each Task	Date: 7/12/88	Number: OPT-12

are worth 0.3. The ratings from groups (b) and (c) are additive. If no conditions pertain for a given task, then a ratings of 0.0 is assigned.

<u>OPT/A21112: Calculate operating-cost savings</u>. This analysis yields an index for operating-cost savings, the second valid reason to use a simulator. These operations-cost savings are associated with replacing some training hours in the actual equipment with training hours in a simulator. The three operations in this process minimize operating cost using a combination of training on actual equipment and training on a defined simulator (OPT/A211121), calculate the operating-cost savings of the minimal cost option over training on actual equipment alone (OPT/A211122), and convert this savings to a savings index (OPT/A211123).

<u>OPT/A21113: Calculate simulation indicator</u>. This activity is accomplished by taking a weighted average of the indices for simulation requirements and operating-cost savings. The user can use the default weights (6B6, set at 0.5 and 0.5) for this calculation or can input his own weights. Since the weights represent the relative importance of the economic and non-economic reasons for simulation, they are subjective and require the considered judgment of the user for their assessment. The equation for this is:



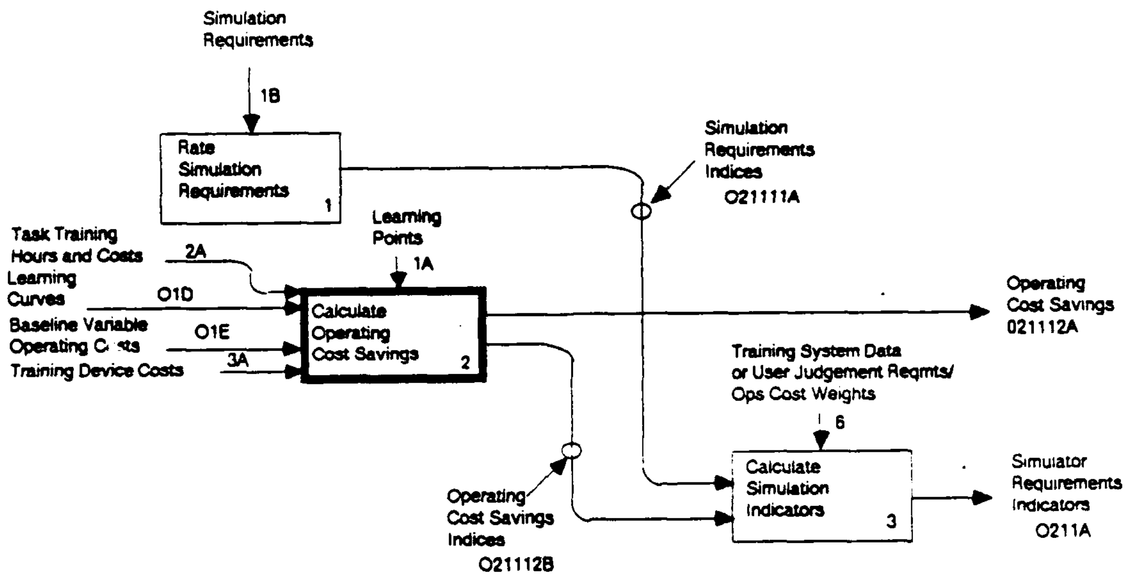
Node: OPT/A2111	Title: Evaluate Simulation Indicators	Date: 7/12/88	Number: OPT-13

$$ISIM_T = WSR * ISR_T + WOCS * IOCS_T,$$

where

$ISIM_T$ = the simulation indicator
WSR = the weight for simulation requirements;
$WOCS$ = the weight for operations-cost savings;
ISR_T = index for simulation requirements;
$IOCS_T$ = index for operations-cost savings.

This equation is overridden if ISR_T equals 1.0, in which case the simulation indicator will be set to 1.0.



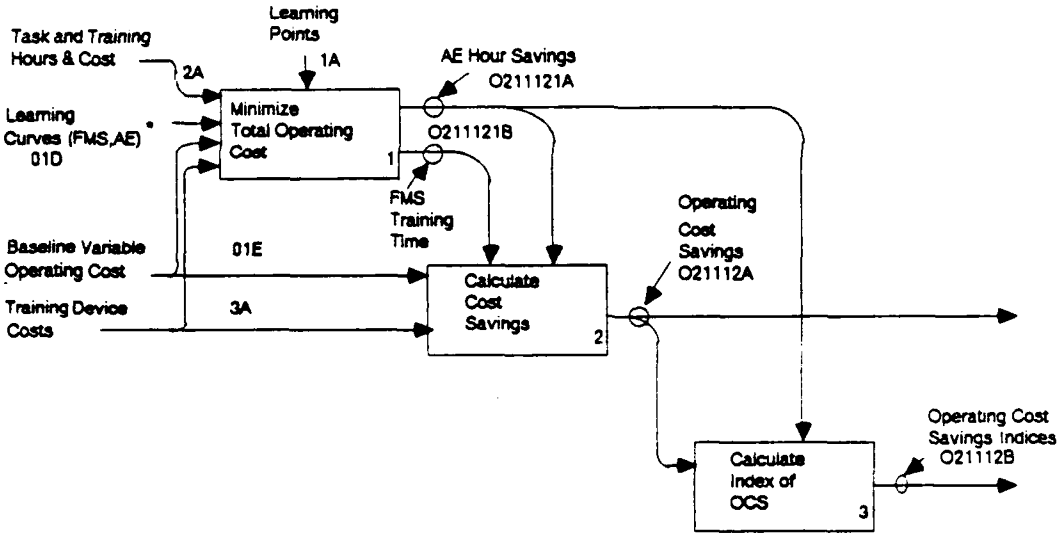
Node: OPT/A2111	Title: Evaluate Simulation Indicators	Date: 7/12/88	Number: OPT-13

OPT/A21112: Calculate Operating-Cost Savings

This process measures the validity of replacing some time on the actual equipment with simulator time for cost reasons.

OPT/A211121: Minimize total operating cost. This analysis uses the learning curves for actual equipment and for an existing FMS (O1D), task entry and standard performance levels (1A), actual equipment (operational and non-operational) hours for training without the simulator (2A), and the hourly operating cost of the FMS (3A) and AE (O1E) to compute the FMS (O211121B) and actual-equipment (O211121A) training times that minimize the total operating cost required to train each task. The cost of training on actual equipment assumes that operational and non-operational hours are reduced in constant proportions by the simulator. The methods described by Cronholm (1985) will be used as the mechanism for this minimization. Both the FMS and actual-equipment learning curves are power functions, and the transfer-of-training function is assumed linear with the multiple equal to the asymptote of the device from which transfer is occurring. Thus, the learning curve has the following form:

$$P_{Ti}(t) = ASM_{Ti}\ (1 - [1 + TM_{Ti}TSF_T(HS_{Ti} + t)]^{-r}),$$



* FMS - Full Mission Simulator. AE - Actual Equipment (e.g., Aircraft)

Node: OPT/A21112	Title: Calculate Operating Cost Savings	Date: 7/12/88	Number: OPT-14

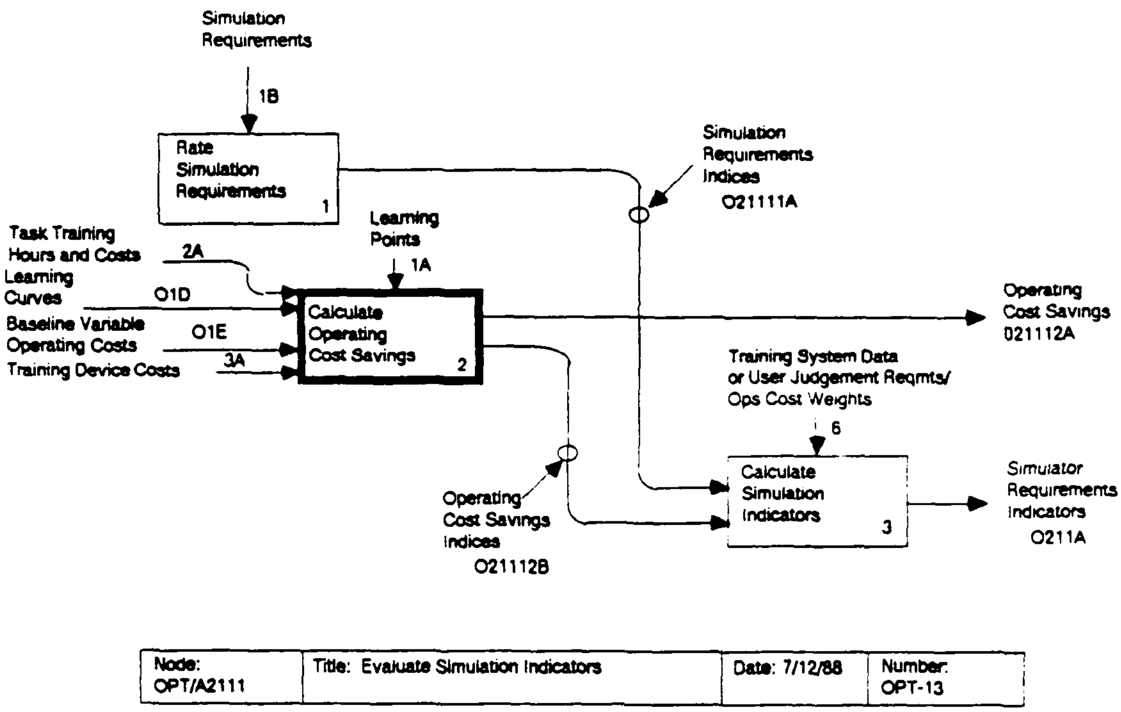
where

P_{Ti} = the level of performance,
t = the training time,
TSF_T = time scaling factor
HS_{Ti} = the effect of previous experience,
TM_{Ti} = time multiplier
r = the learning rate (assumed to be 0.7), and
ASM_{Ti} = the asymptote of the learning curve, a measure of transfer of training.

Cronholm (1985) showed that the derivative of training performance on the AE with respect to training time on the FMS must be equal to the FMS cost rate divided by the AE cost rate to minimize total operating cost. Using the above learning curve, this translates to:

$$\frac{VHRC_{FMS}}{VHRC_{T,AE}} = \left[1 - \frac{ASM_{FMS}}{ASM_{AE}}\left[1-[1+TM_{FMS}TSF\ (HS_{FMS}+t_{FMS})]^{-r_{FMS}}\right]\right]^{-\frac{1}{r_{AE}}-1}$$

$$* \frac{TM_{FMS}}{TM_{AE}} * \frac{ASM_{FMS}}{ASM_{AE}} * [1+TM_{FMS}TSF\ (HS_{FMS}+t_{FMS})]^{-r_{FMS}-1}$$



where

P_{FMS} = $ASM_{FMS} * [1-(1+TM_{FMS}TSF(HS_{FMS}+t_{FMS}))^{-r_{FMS}}]$ is the FMS learning curve,

P_{AE} = $ASM_{AE} * [1-(1+TM_{AE}TSF(HS_{AE}+t_{AE}))^{-r_{AE}}]$ is the AE learning curve

$VHRC_{FMS}$ = hourly cost of FMS

$VHRC_{T,AE}$ = hourly cost of AE

The subscript T indicating the task is assumed throughout this section. t_{FMS} is solved for as follows:

$$t_{FMS} = \frac{1}{TM_{FMS}\ TSF} [(1 - PX/ASM_{FMS})^{-1/r} - (1 - PENTRY/ASM_{FMS})^{-1/r}]$$

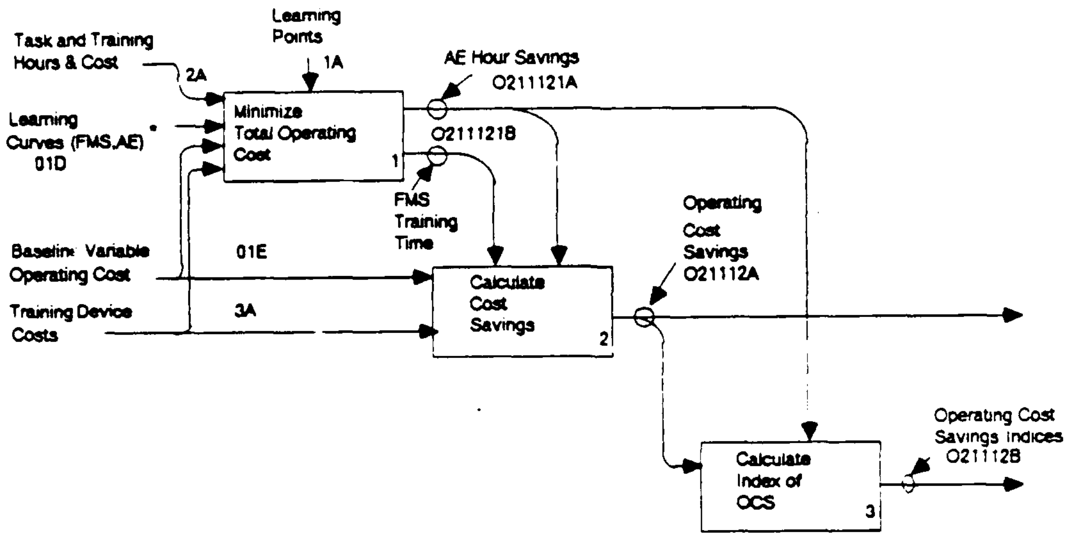
where

ENT = the entry level performance for task T, and

PX = the crossover performance level between the FMS and the AE, calculated by the following equation.

$$PX = \frac{ASM_{AE} - (ASM_{FMS} * z)}{1 - z}$$

where



* FMS - Full Mission Simulator. AE - Actual Equipment (e.g., Aircraft)

Node: OPT/A21112	Title: Calculate Operating Cost Savings	Date: 7/12/88	Number: OPT-14

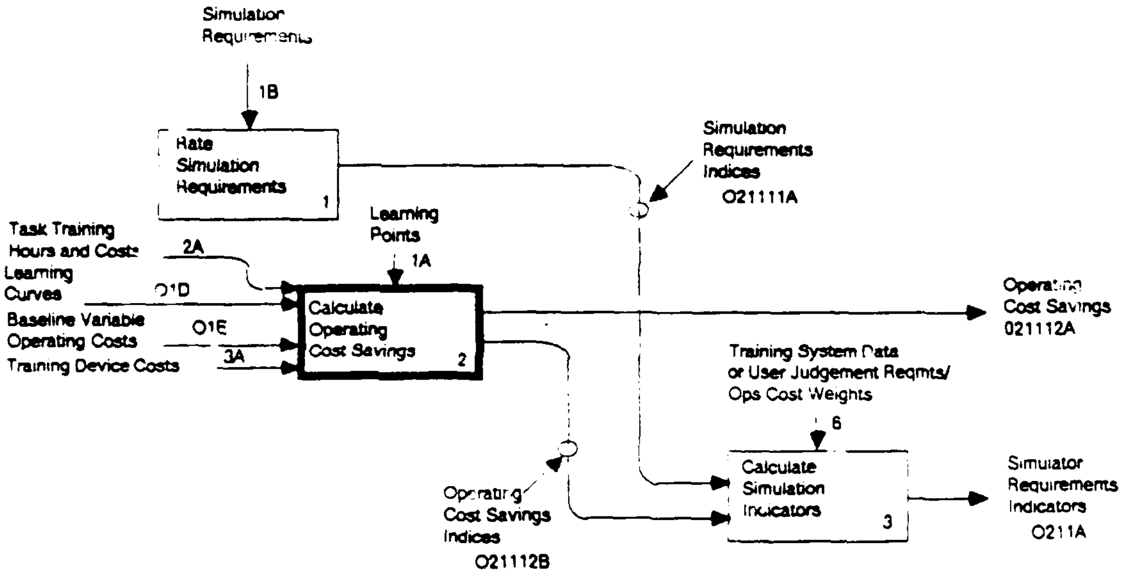
$$z = \frac{ASM_{AE}}{ASM_{FMS}} * \left[\frac{ASM_{FMS}}{ASM_{AE}} * \frac{TM_{FMS}}{TM_{AE}} * \frac{VHRC_{T,AE}}{VHRC_{FMS}} \right]^{r/(r+1)}$$

t_{AE} is calculated as follows:

$$t_{AE} = \frac{1}{TM_{AE}\ TSF} \left[(1 - STD/ASM_{AE})^{-1/r} - (1 - PX/ASM_{AE})^{-1/r} \right]$$

OPT/A211122: Calculate cost savings. This is a straightforward operation using the results of the previous process. The operating cost without a simulator is $CVAR_T$ (O13C). This minimal cost is the simulator cost (hours times operating cost) plus the reduced actual equipment operating cost. The cost savings (OCS_T; O21112A), then, is the difference between the minimal cost and the cost of training without a simulator.

OPT/A211123: Calculate index of operating-cost savings. This analysis uses the actual-equipment cost savings of all tasks to compute a normalized index, such that the task with the greatest operating-cost savings receives an index value of 1:



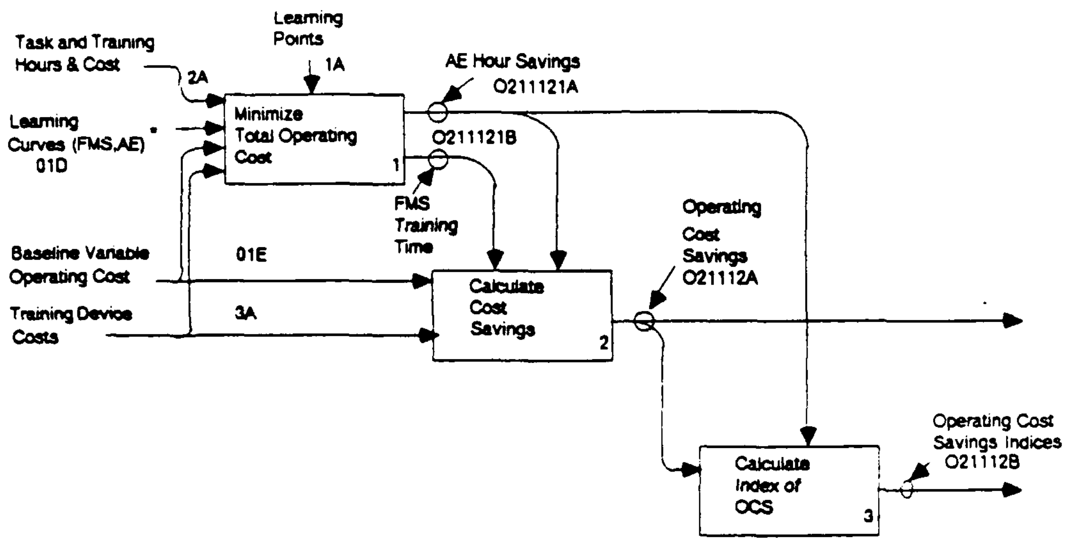
Node: OPT/A2111	Title: Evaluate Simulation Indicators	Date: 7/12/88	Number: OPT-13

$$IOCS_T = OCS_T / \max_T(OCS_T)$$

where

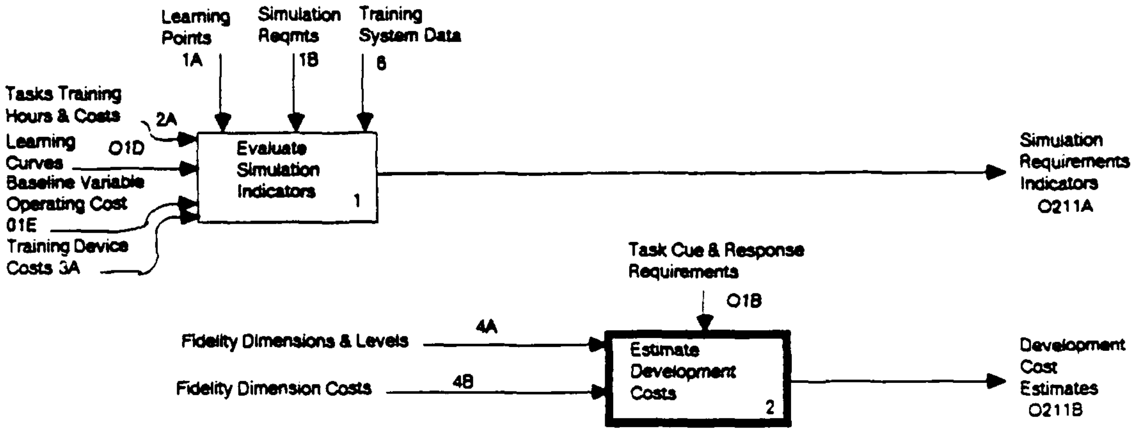
$IOCS_T$ = the operating-cost savings index for task T, and
OCS_T = the operating-cost savings for task T.

This index is guaranteed to be between 0.0 and 1.0.



* FMS - Full Mission Simulator. AE - Actual Equipment (e.g., Aircraft)

Node: OPT/A21112	Title: Calculate Operating Cost Savings	Date: 7/12/88	Number: OPT-14



Node: OPT/A211	Title: Evaluate Each Task	Date: 7/12/88	Number: OPT-12

OPT/A2112: Estimate Development Costs

This process uses the task cue and response requirements to estimate the development cost of a training device that could provide training to meet the performance standard for task.

OPT/A21121: Calculate Fidelity Dimension Costs. This procedure uses a technical performance index to estimate a cost equation for fidelity dimension j using the following equations:

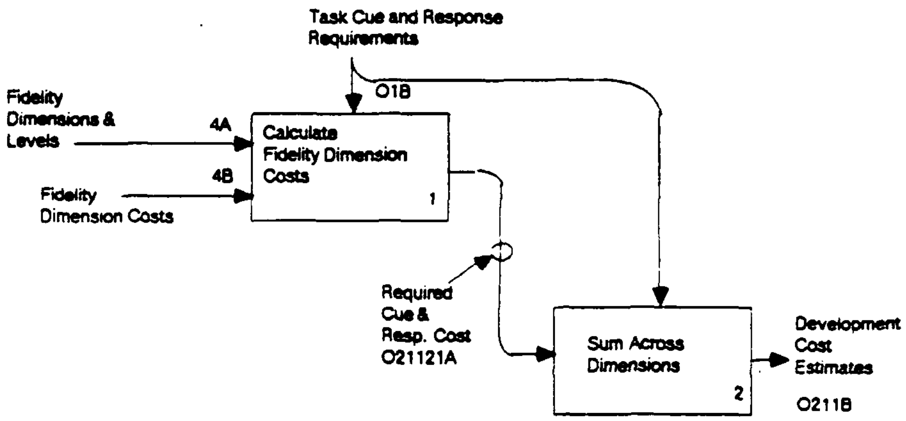
$$DCST_{Tj}(FRQT_{Tj}) = CMIN_j + (CMAX_j - CMIN_j)$$
$$* \left[-(1/E_j)\ \ln\left(1 - \left[\frac{FRQT_{Tj} - TPMIN_j}{TPMAX_j - TPMIN_j}\right](1 - e^{-E_j})\right)\right]$$

where

$TPMIN_j$ = Technical performance of the least capable FMS option for cue/response dimension j

$TPMAX_j$ = Technical performance of the most capable FMS option for cue/response dimension j

$CMIN_j$ = development cost of the minimum technical performance index on fidelity dimension j,



Node: OPT/A2112	Title: Estimate Development Costs	Date: 7/12/88	Number: OPT-15

$CMAX_j$ = development cost of the maximum technical performance index (set to 1.0) on fidelity dimension j,

E_j = empirically determined cost equation parameter for fidelity dimension j,

$FRQT_{Tj}$ = cue and response requirement for task T on fidelity dimension j.

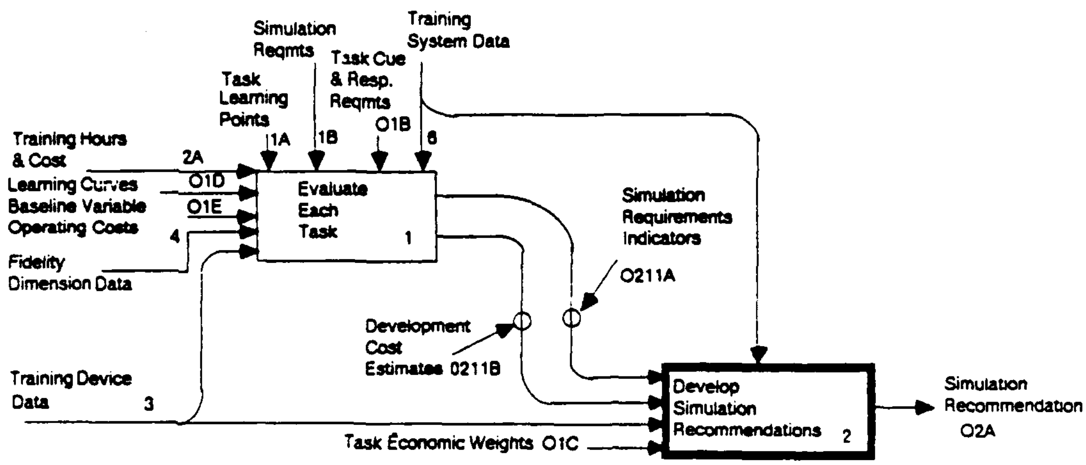
If $FRQT_{Tj}$ is less than $TPMIN_j$, the performance parameter associated with $CMIN_j$, then

$$DCST_{Tj}(FRQT_{Tj}) = \frac{FRQT_{Tj}}{TPMIN_j} \times CMIN_j.$$

<u>OPT/A21122: Sum Across Dimensions</u>. The development cost for a given task is the sum of the fidelity-dimension estimates. The index for development cost estimates is the normalization of the costs so that the task with the highest development cost has an index value of 1.0 and the others are scaled accordingly.

$$DEVC_T = \sum_j DCST_{Tj}$$

$$IDEVC_T = DEVC_T/(\max_T(DEVC_T))$$



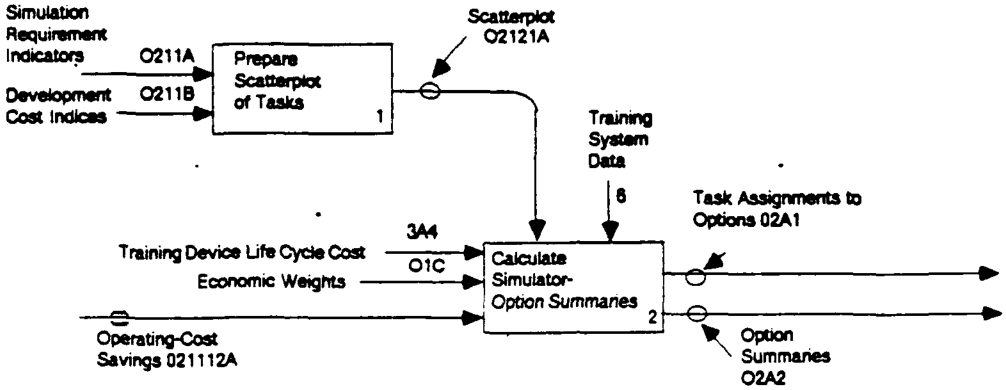
Node: OPT/A21	Title: Recommend Simulator Configuration	Date: 7/12/88	Number: OPT-11

OPT/A212: Develop Simulation Recommendations

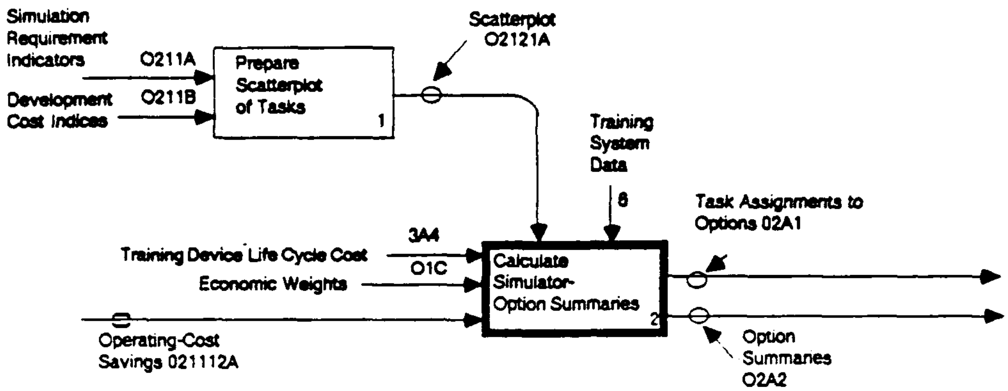
This process uses the task indicators for simulation and estimated development cost indices to formulate a recommendation for a full-mission simulator (FMS), one or more part-mission simulators (PMS), or use of actual equipment (AE).

OPT/A2121: Prepare scatterplot of tasks. This is a process that produces an x,y plot of each task; that is, the simulation requirement indicator (O211A) is plotted along the y-axis and the development cost index (O211B) is plotted along the x-axis for each task.

OPT/A2122: Calculate simulator-option summaries. This analysis uses the scatterplot from the previous process to assign tasks to recommendation regions and then compute summaries for the FMS, PMS, and AE for the options within the scope of this analysis. In the three operations included in this activity, the cost boundary between the AE region and the FMS and PMS regions is determined (OPT/A21221), tasks are assigned to the appropriate regions (OPT/A21222), and summary values are calculated for the three regions (OPT/A21223).



Node: OPT/A212	Title: Develop Simulation Recommendations	Date: 7/12/88	Number: OPT-16



Node: OPT/A212	Title: Develop Simulation Recommendations	Date: 7/12/88	Number: OPT-16

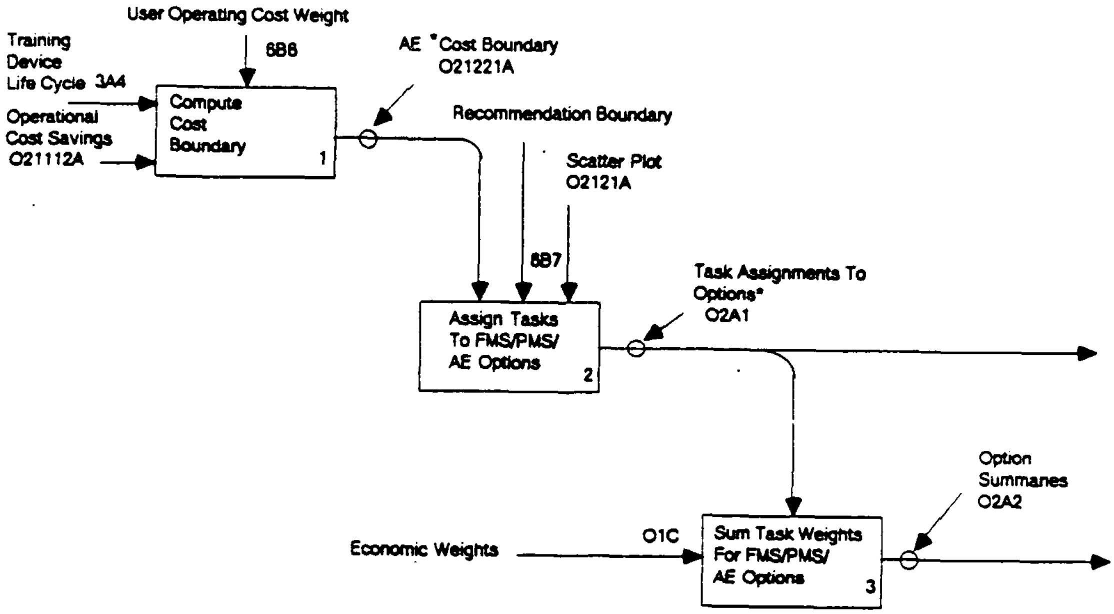
OPT/A2122: Calculate Simulator-Option Summaries

The option summaries that result from this process indicate the relative economic and training improvement-based effects that an FMS or PMS might have. If these numerical summaries are high enough for either or both of the FMS or PMS options, then the OSBATS recommendation will be favorable.

OPT/A21221: Compute cost boundary. This process uses data about operating-cost savings of simulation (O21112A1), the life cycle of the simulator (3A4), and the weight given operating cost savings index (6B6) in determining the indicator for simulation requirements to determine the cost trade-off for recommending a simulator. The major assumption made in computing this cost boundary is that if a given task has no requirements for simulation, the decision to use a simulator is based on economics only (operating-cost savings versus development cost). Simulation requirements provide additional benefits of simulation that are added to the economic benefits. The following three equations are the basis of this computation:

(1) $ISIM = IOCS * WOCS$ by assumption

(2) $IOCS = OCS / OCS_{max}$ by definition



* FMS - Full Mission Simulator, PMS - Part-Mission Simulator, AE - Actual Equipment

Node: OPT/A2122	Title: Calculate Simulator Option Summaries	Date: 7/12/88	Number: OPT-17

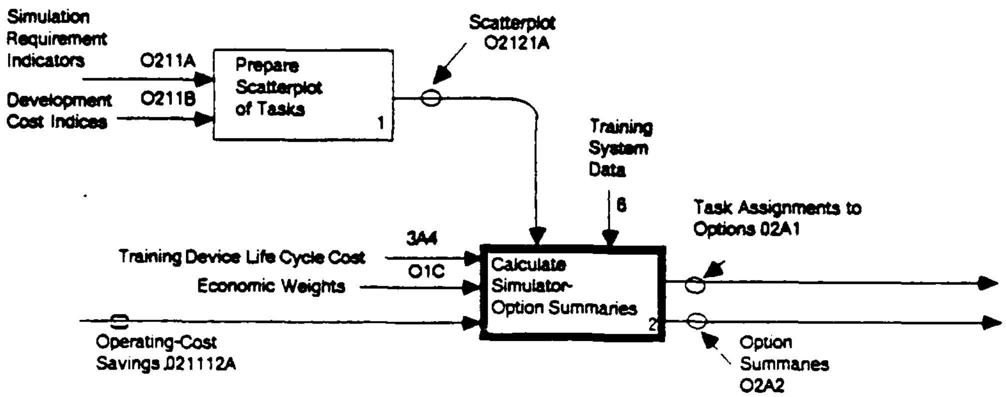
(3) $DEVC = OCS_{max} * PB * TSIM * GRADS$ cost accounting

where

- IOCS = index of operations cost savings;
- OCS = operations-cost saving of the simulator (O21112A);
- ISIM = indicator for simulation;
- WOCS = weight for operations cost savings in calculation of ISIM (6B6);
- DEVC = development cost of a simulator;
- PB = payback period needed for the simulator (3A4);
- TSIM = number of tasks over which the simulator development cost should be amortized;
- GRADS = the required number of graduates per year (6A1).

The first equation embodies the assumption that there are no requirements for simulation. The second equation is the definition of the operating-cost savings index. The third equation expresses the criterion for the cost boundary that the training device development cost is saved over the payback period (3A4) for all tasks for which the device is used (tasks with positive cost savings). By solving for ISIM as a function of DEVC we find

$$ISIM = WOCS/(PB * TSIM * OCS_{max} * GRADS) * DEVC$$



Node: OPT/A212	Title: Develop Simulation Recommendations	Date: 7/12/88	Number: OPT-16

However, since (TSIM * OCS_{max}) overapproximates the savings due to the simulator, we are substituting (from OPT/A211122):

$\sum_{T} OCS_T$.

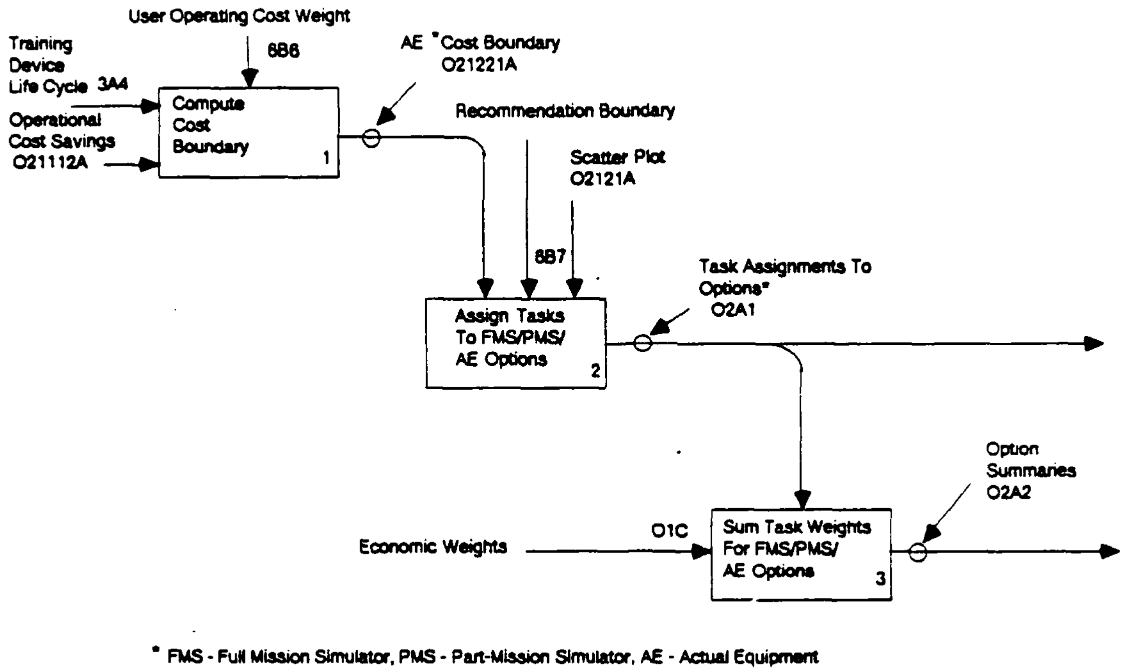
So we have a linear cost boundary with slope,

$WOCS/(PB * \sum_{T} OCS_T * GRADS)$,

relating the x and y dimensions of the scatterplot. Any task lying above this line would qualify for a simulator; and those below would not. User adjustments in the cost boundary are allowed due to the uncertainty in the actual cost savings that would result from simulation.

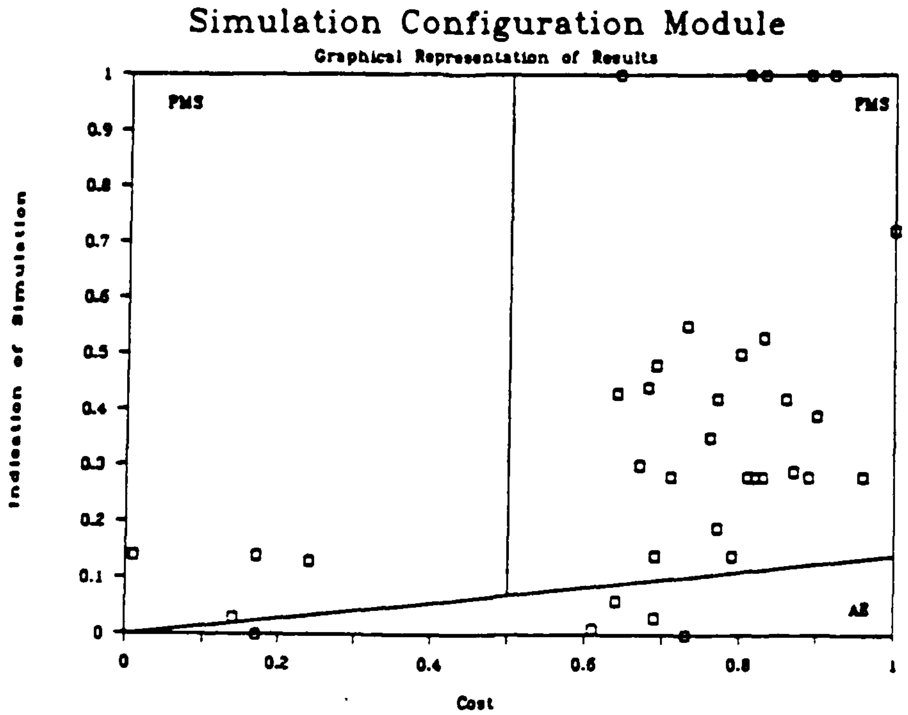
<u>OPT/A21222: Assign tasks to FMS, PMS and NoS options</u>. The linear cost boundary and a vertical boundary for a specified value of the fidelity index that separates a FMS from PMSs are used to assign tasks in the scatterplot to FMS, PMS or AE options. (See the figure opposite.)

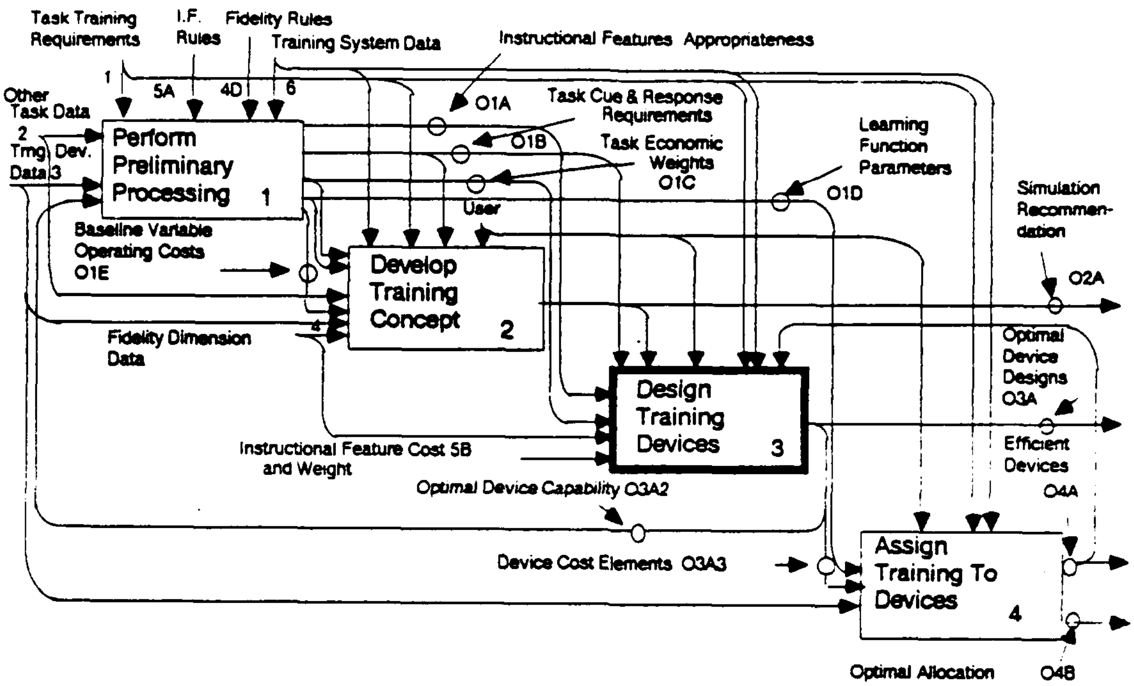
<u>OPT/A21223: Sum task weights for FMS, PMS, and NoS options</u>. The task weights (O1C) of all tasks in the FMS region



Node: OPT/A2122	Title: Calculate Simulator Option Summaries	Date: 7/12/88	Number: OPT-17

of the scatterplot are summed to be the FMS summary. The same is done for the PMS tasks and the AE tasks. These summary values are displayed to the user.





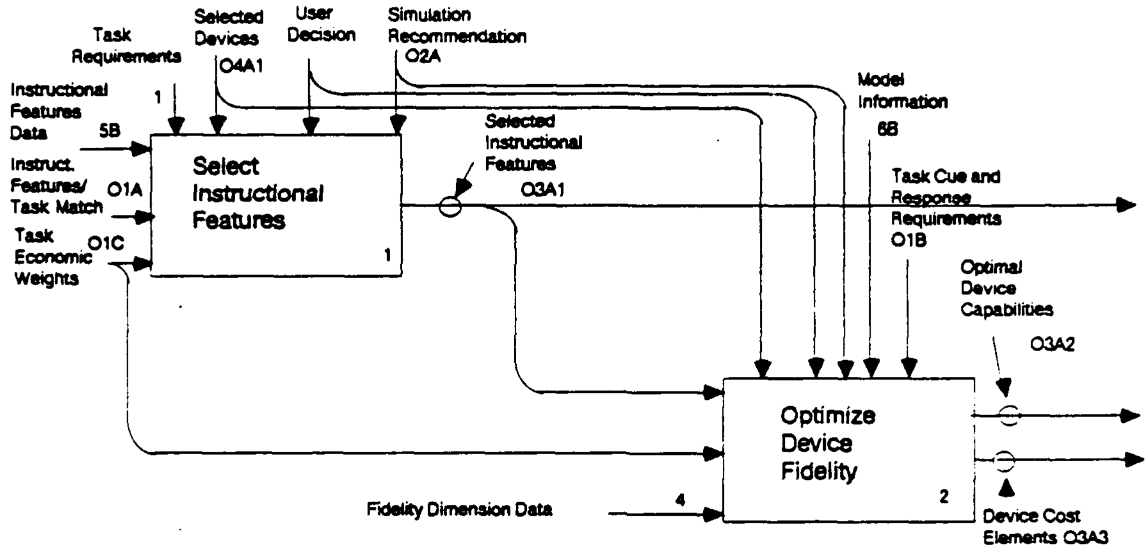
Node: OPT/A0	Title: Optimize Training System Design	Date: 7/27/88	Number: OPT-2

OPT/A3: Design training devices.

Two tools are used to aid in the design of training devices. The first tool determines the cost-effective instructional features to be incorporated into a training device. The second tool determines the device areas in which technical sophistication and fidelity would be cost-effective.

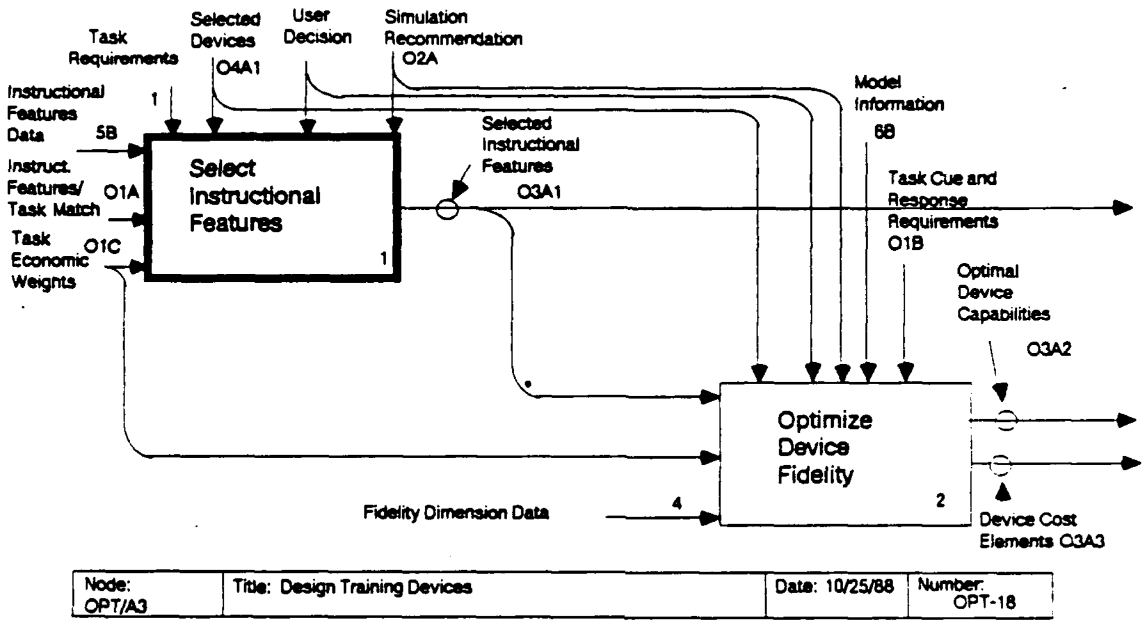
OPT/A31: Select instructional features. This tool examines the instructional features that would improve training efficiency for each task (O1A). Then the tool identifies the instructional features that have the greatest expected impact on training efficiency, given their cost. The output of this module is a list of the instructional features, ordered by decreasing benefit/cost ratio. Features at the beginning of the list provide a better value, given their development cost, than the features at the end of the list. The optimal set of instructional features at any budget may be determined by selecting features in order of decreasing benefit/cost ratio until the budget is met. The results of this model may be integrated into the analysis performed by the fidelity optimization model.

OPT/A32: Optimize device fidelity. This tool addresses the problem of how much should be invested in the technical



Node: OPT/A3	Title: Design Training Devices	Date: 10/25/88	Number: OPT-18

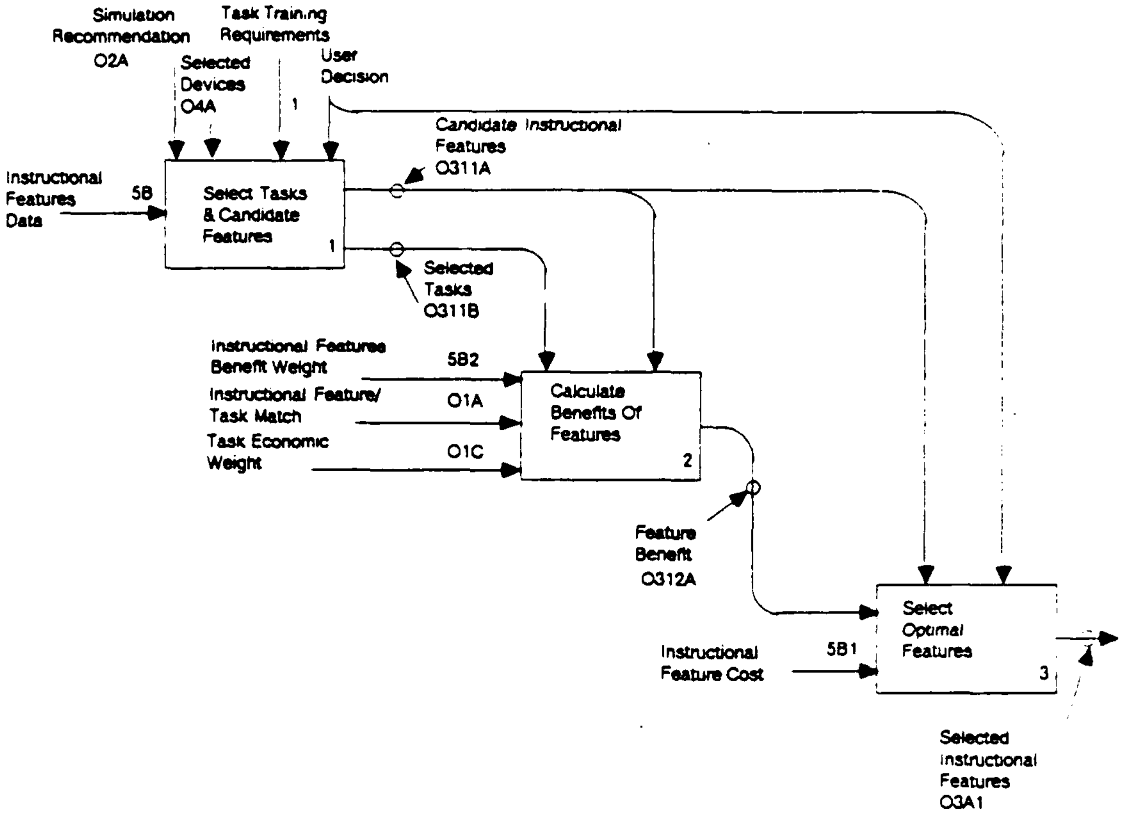
sophistication and fidelity of a training device. The tool considers several dimensions of fidelity that are related to task cue and response requirements. A training device may be more or less sophisticated on each fidelity dimension. The module compares task requirements on these dimensions with the cost of meeting these requirements, in order to determine the areas in which an investment in increased fidelity can be justified by increased training effectiveness. The output of this model is a collection of training device designs that optimize training effectiveness as a function of investment cost. Given this set, the optimal training-device design at any development cost may be determined. The choice of the single training device design that meets training criteria at the lowest cost from the set of optimal device designs should be made by either the Training Device Selection Module or the Resource Allocation Module.



OPT/A31: Select Instructional Features

This tool for designing training devices identifies, given a development cost, the instructional features that will improve training for the most tasks. Instructional features are presumed to influence the efficiency of training rather than the maximum effect of training or transfer of training. Thus, instructional features influence the time multiplier in the learning model rather than the asymptote. Three major activities compose this tool: candidate features are selected from a master list, the benefits of these features are determined, and the features that provide the greatest benefit for the cost are determined. The outputs of this module may serve as inputs to the Fidelity Optimization Module.

OPT/A311: Select tasks and candidate features. In this activity, the user selects the training tasks to be used as the basis for the evaluation (O311B). This selection is described in node OPT/A3111. The selected tasks may be one of the clusters produced by the Simulation Configuration Module (O2A), the tasks trained on one device as determined by the Training Device Selection Module (O4A), or a group of tasks selected by the user. In addition, the user selects the instructional features to be evaluated in the analysis (O311A) from a master list of instructional features (5B). This selection is described in

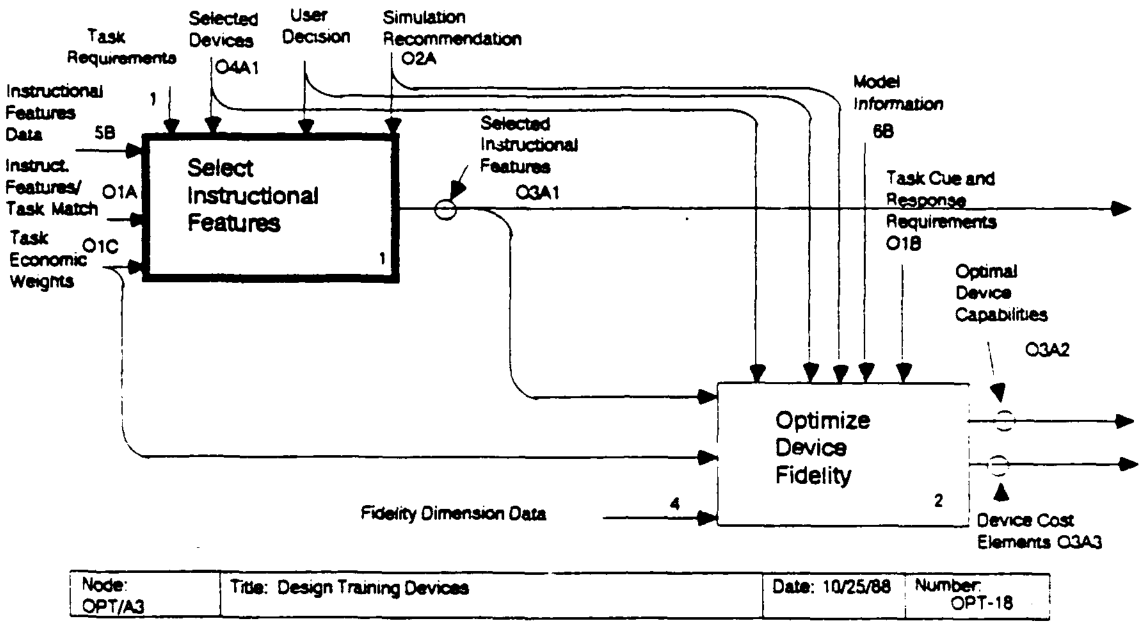


Node: OPT/A31	Title: Select Instructional Features	Date: 7/12/88	Number: OPT-19

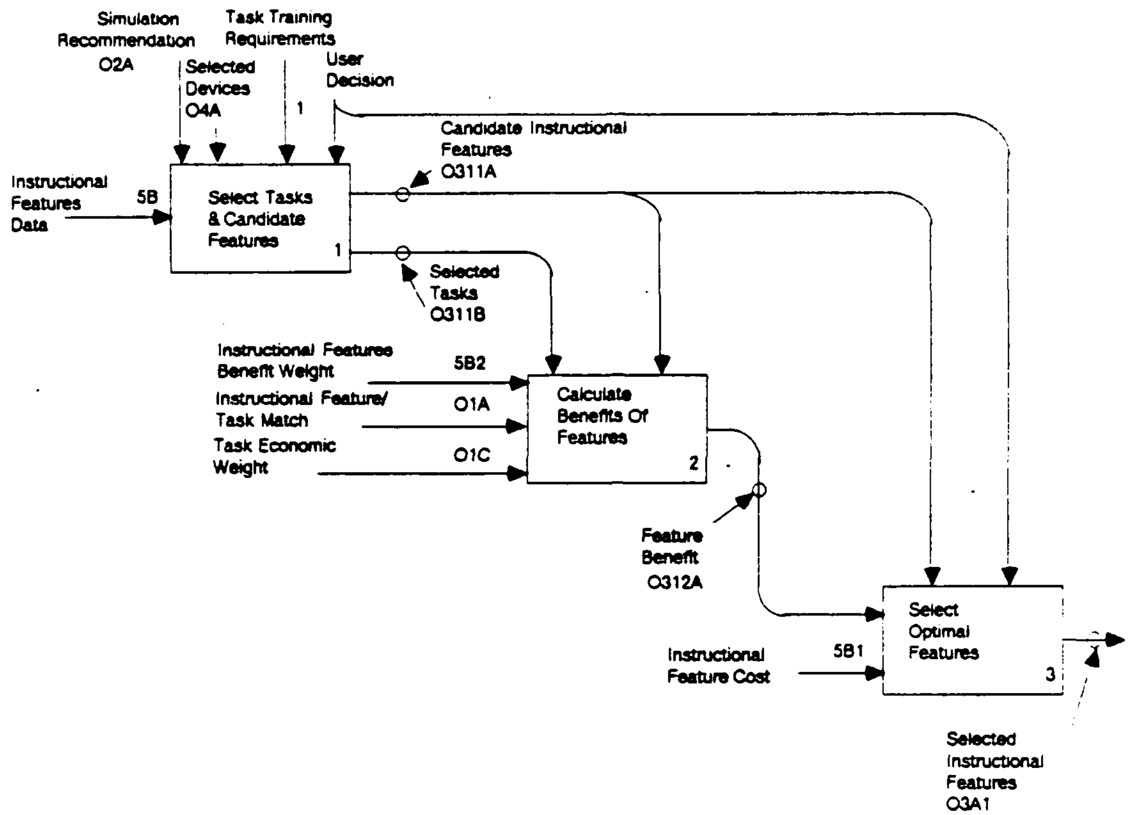
node OPT/A3112. Specific procedures for this activity are described in the detailed diagram for this node.

<u>OPT/A312: Calculate benefits of features</u>. In this activity, the benefits of the candidate instructional features (O312A) are determined. In order to determine instructional feature benefit, the module must aggregate (1) the number of tasks for which the instructional feature is relevant (O1A), (2) the economic costs of training these tasks on actual equipment (O1C), and (3) the likelihood that the instructional features will be used by instructors (5B2). The overall benefit of an instructional feature is the product of these three factors. The subactivities of this activity first calculate a measure of feature effectiveness that incorporates the first two factors described above (OPT/A3121), and then incorporate the third factor (OPT/A3122).

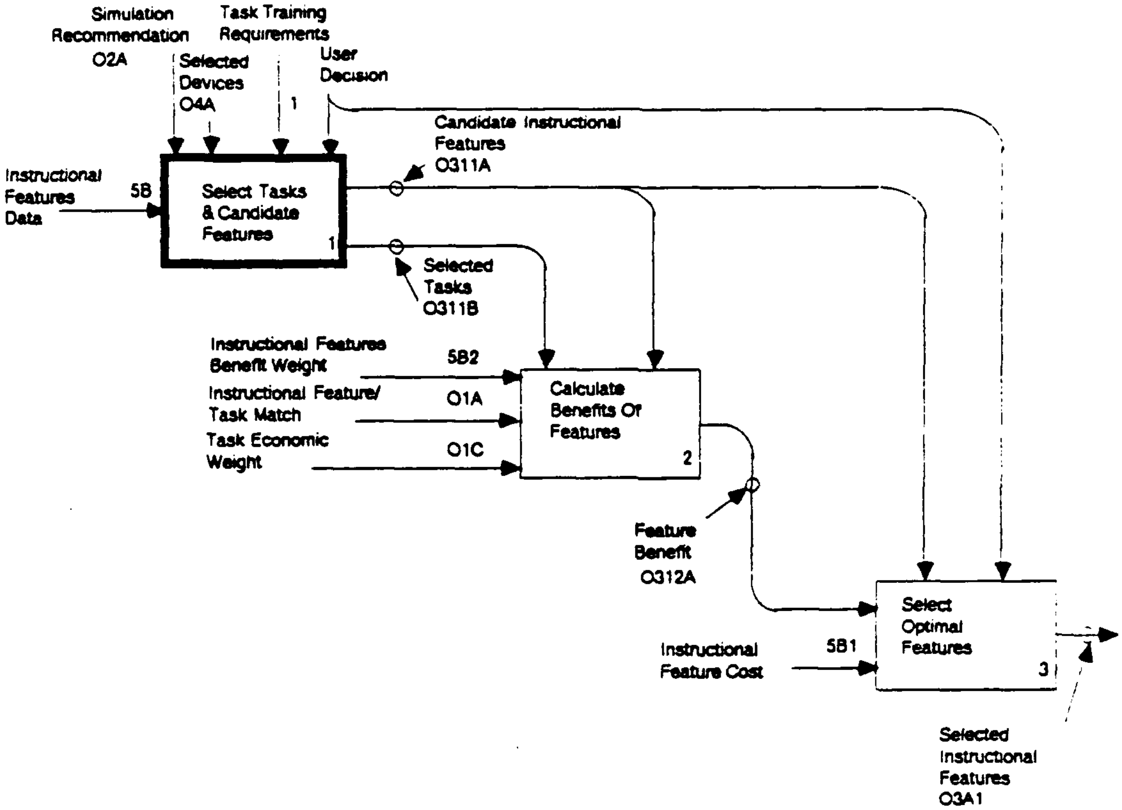
<u>OPT/A313: Select optimal order of features</u>. In this activity, the instructional-feature benefit measures (O312A) are combined with assessments of feature-investment cost (5B1), and the ratio of benefit to cost is used to determine the optimal selection of instructional features at any cost (O3A1). The user may examine the optimal selection of features at any development cost. In addition, the entire ordered list of instructional features, or some portion of it, may serve as input to the



fidelity optimization module. The optimal order of features is determined by computing the benefit/cost ratio for each feature (OPT/A3131), sorting the features by benefit/cost ratio (OPT/A3132), and listing the instructional features (OPT/A3133).



Node: OPT/A31	Title: Select Instructional Features	Date: 7/12/88	Number: OPT-19



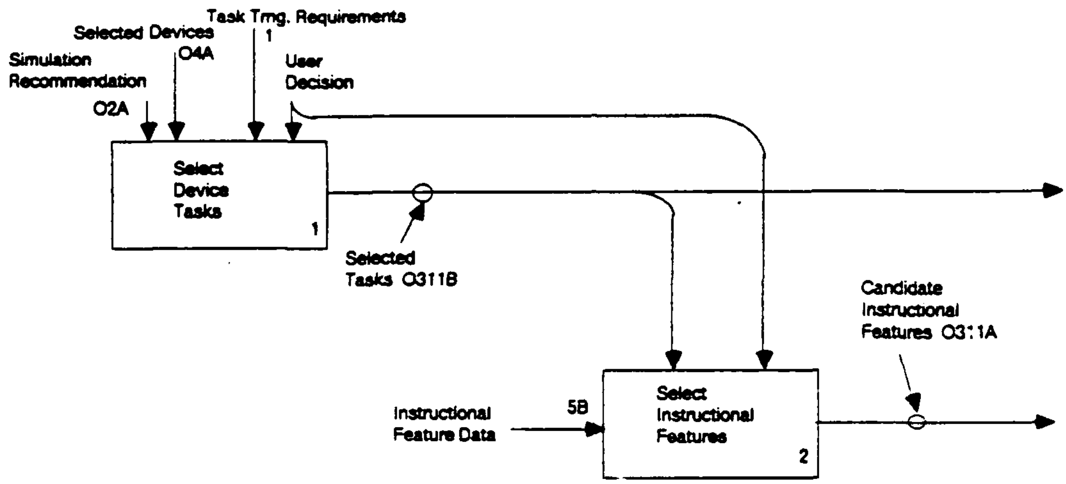
Node: OPT/A31	Title: Select Instructional Features	Date: 7/12/88	Number: OPT-19

OPT/A311: Select Tasks and Candidate Features

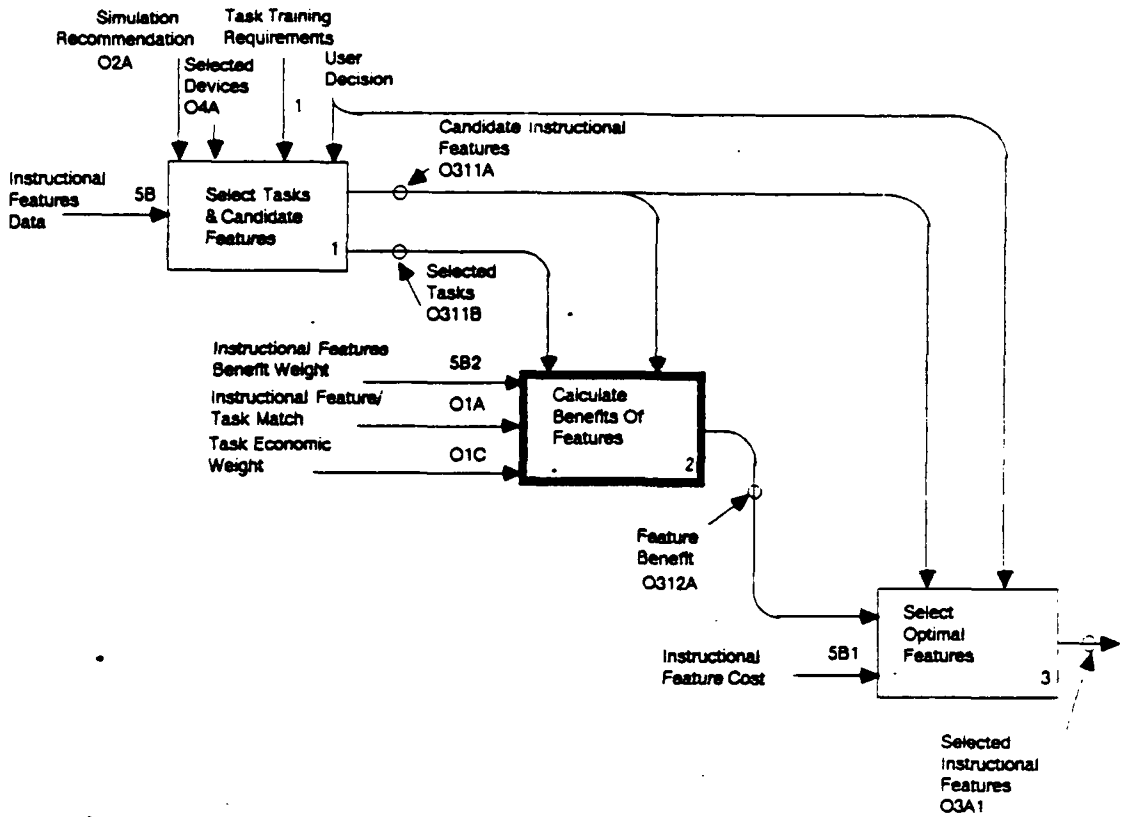
The data base contains a master list of instructional features. The user may not want to consider all features on the list. Consequently, this activity gives the user an opportunity to modify the list of instructional features. In addition, the user is given an opportunity to select the tasks that are to be used to evaluate the candidate instructional features.

OPT/A3111: Select device tasks. The tasks included in the analysis define the requirements for the training device being designed. The optimal instructional features for the training device depend on these training requirements. The Simulation Configuration Module (O2A) and Training Device Selection module (O4A) both make recommendations regarding the tasks assigned to a training device. The user may use the recommendations of one of these models, or may select other tasks to be used in determining instructional-feature benefit.

OPT/A3112: Select instructional features. In this step, the user selects the instructional features to be evaluated in the analysis from a master list of possible features (5B). In this way, the user may eliminate from consideration those features that are infeasible for reasons outside the scope of the model.



Node: OPT/A311	Title: Select Tasks and Candidate Features	Date: 7/12/88	Number: OPT-20



Node: OPT/A31	Title: Select Instructional Features	Date: 7/12/88	Number: OPT-19

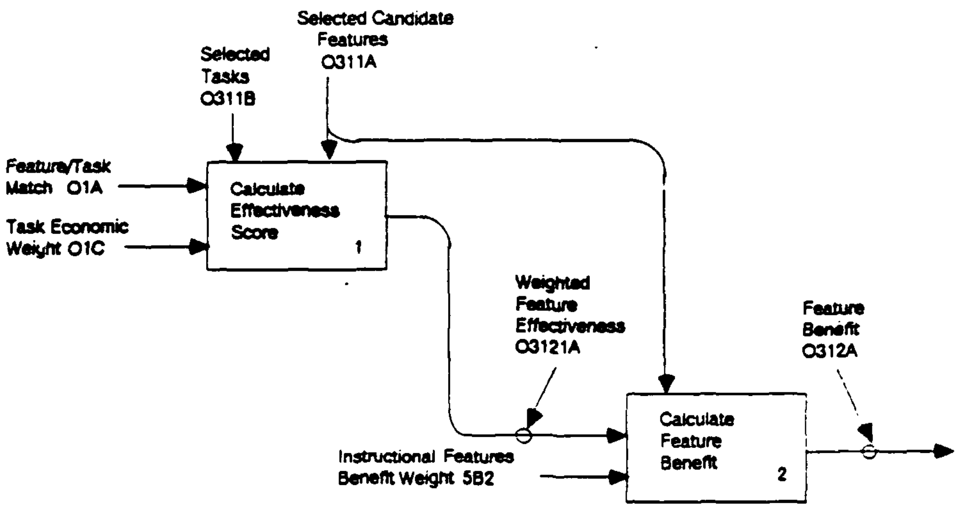
<u>OPT/A312: Calculate Benefits of Features</u>

An instructional feature's benefit depends primarily on the number of tasks for which the feature enhances training efficiency. In aggregating over tasks, tasks that are expensive to train on actual equipment are given higher weight than tasks that are relatively inexpensive to train. The task weight (O1C) is a measure of the costs to train the tasks on actual equipment. A feature's overall effectiveness measure is multiplied by a benefit weight that represents the likelihood that a feature would actually be used by an instructor.

<u>OPT/A3121: Calculate effectiveness score</u>. The effectiveness score (EFF_k; O3121A) aggregates the feature/task match matrix (IFT_{Tk}) over tasks. The overall score is a weighted sum of IFT_{Tk} (O1A), with weights being the economic weights (TWT_T) of the tasks (O1C). The effectiveness score (EFF_k) is given by the following equation:

$$EFF_k = \sum_T (IFT_{Tk} \times TWT_T)$$

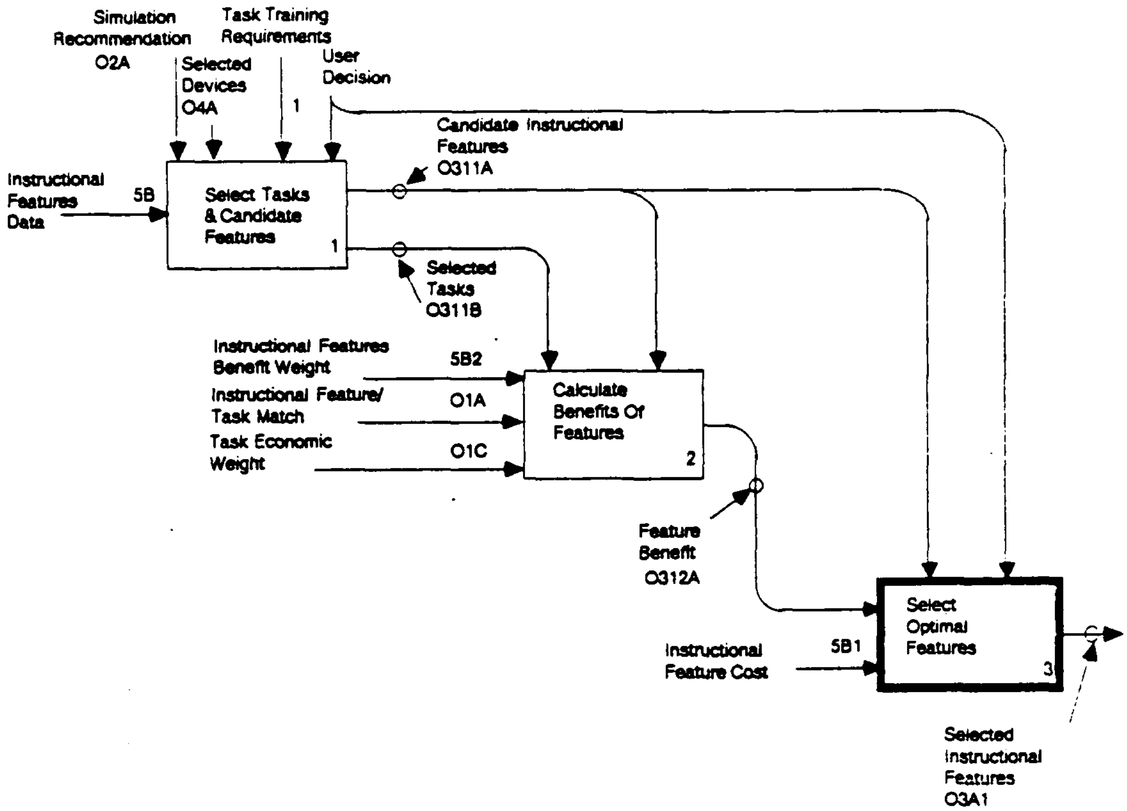
<u>OPT/A3122: Calculate feature benefit</u>. Recent analyses (Logicon, 1985) suggest that an evaluation of instructional features consider overall feature benefit in addition to the



Node: OPT/A312	Title: Calculate Benefits of Features	Date: 7/12/88	Number: OPT-21

task-specific measures (O3121A) described above. This benefit accounts for frequency of need, instructor loading, and feature useability. We have implemented these suggestions by incorporating an instructional-feature benefit weight ($IFWT_k$). The benefit weight (5B2) summarizes historical data on the likelihood that an instructional feature would be used. The overall benefit (BEN_k) for instructional feature k is given by:

$$BEN_k = EFF_k \times IFWT_k$$



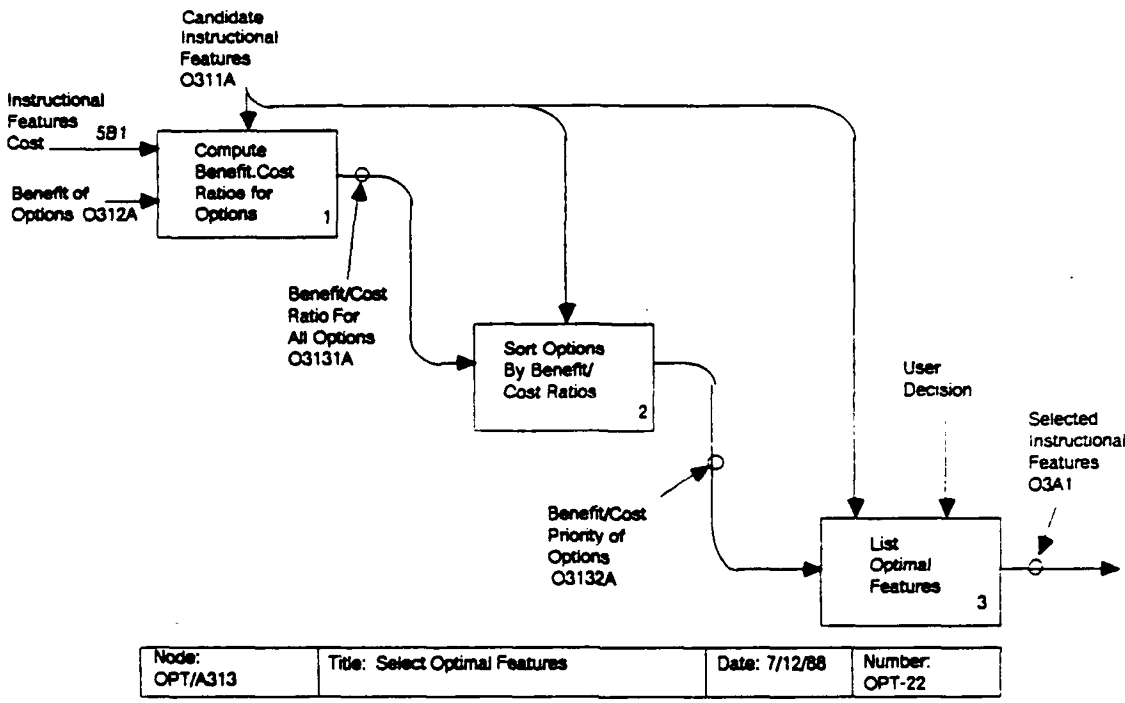
Node: OPT/A31	Title: Select Instructional Features	Date: 7/12/88	Number: OPT-19

OPT/A313: Select Optimal Features

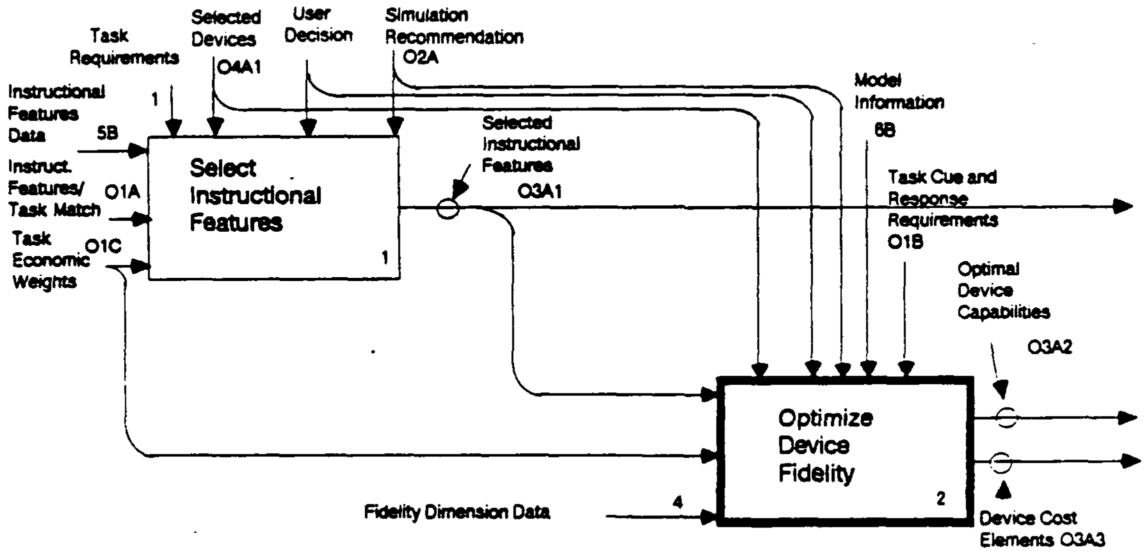
This process combines the calculated instructional-feature benefits (O312A) with assessed feature costs (5B1). It determines the optimal features to be included in the design of a training device as a function of the investment cost budget for instructional features. The user may select one or more options to be evaluated in the Fidelity Optimization Module.

OPT/A3131: Compute benefit/cost ratios for options. The k-th instructional feature is represented by both a benefit (BEN_k; O312A) and a development cost (IFC_k; 5B1). This activity divides the benefit by the cost to obtain a benefit/cost ratio (BEN_k/IFC_k).

OPT/A3132: Sort options by benefit/cost ratio. This activity orders the features according to decreasing benefit/cost ratio. If the features being considered are presumed independent, they can be properly prioritized on the basis of BEN_k/IFC_k. The assumption of independence deserves some investigation, particularly in terms of its interaction with the assumptions used to estimate the effect of instructional features in the Training Device Selection Module.



<u>OPT/A3133: List optimal features</u>. This activity presents the results of the analysis so that the user may determine the instructional features that should be included to meet cost or benefit criteria. The optimal list of features at any cost (or benefit) level is found by adding features to the list in sorted order until the cumulative cost (or benefit) is equal to the criterion level. This activity takes the range of costs and produces as output the set of optimal instructional-feature packages for all costs in the range. Included in the output are the cost and benefit for each package.

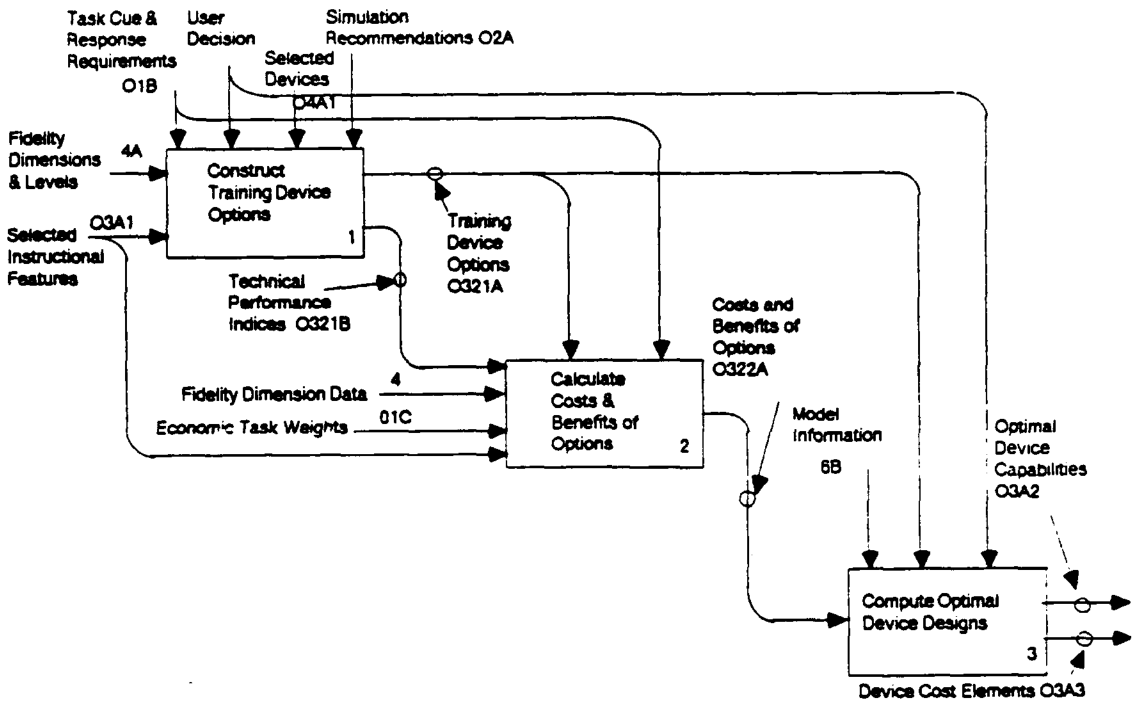


Node: OPT/A3	Title: Design Training Devices	Date: 10/25/88	Number: OPT-18

<u>OPT/A32: Optimize Device Fidelity</u>

The goal of this tool is to consolidate task cue and response requirements for the tasks being considered for training on a training device with training device fidelity options so that optimal device designs can be identified.

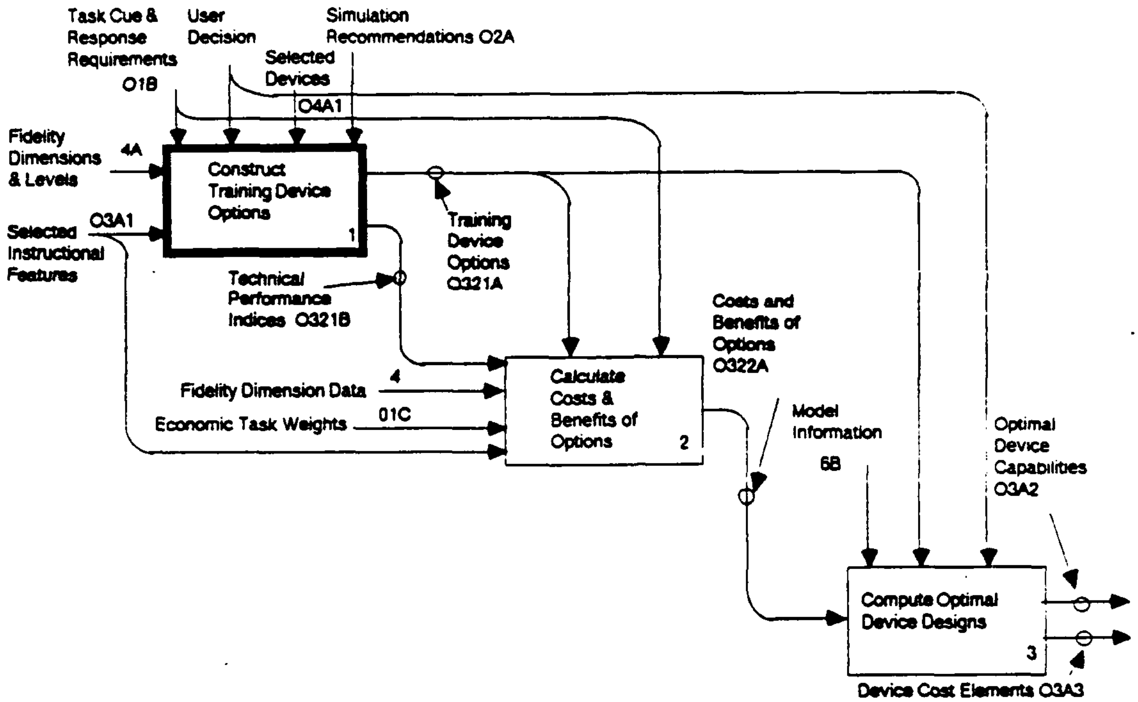
<u>OPT/A321: Construct training device options</u>. This activity allows the user to examine the task cue and response requirements (O1B) of the tasks specified in the training concept (O2A) to select relevant fidelity dimensions (in OPT/A3211) and pick the minimum and maximum options in each fidelity dimension (in OPT/A3212) from a master list of options in the data base (4A). The user is also asked to determine whether the results of the instructional features module, if available, should be incorporated as a fidelity dimension in this module (OPT/A3213). Finally the user may discard any options between the minimum and maximum for a given fidelity dimension (OPT/A3214). These options and dimensions have been designed to be as independent of each other as possible--independent in the sense that cost and benefit of each option do not depend on the options chosen for other dimensions. The outputs of this activity are a set of candidate training device options (O321A) and associated technical performance indices (O321B) from the data base (4A3).



Node: OPT/A32	Title: Optimize Device Fidelity	Date: 7/12/88	Number: OPT-23

<u>OPT/A322: Calculate costs and benefits of options</u>. This process has four operations that establish development costs and training benefits for each option. These costs and benefits are comparable across all fidelity dimensions and are based upon the technical performance indices for each option. Costs are determined from technical performance by a logarithmic estimation function in the first subactivity (OPT/A3221). The remaining three subactivities determine benefit by determining task trainability (OPT/A3222), calculating a benefit score within each fidelity dimension (OPT/A3223), and determining weights that place benefit on a common scale across dimensions (OPT/A3224).

<u>OPT/A323: Compute optimal device designs</u>. This process has four operations. The optimal training device designs are based upon the incremental benefit-to-cost ratios of the options. After these ratios are computed (OPT/A3231) the options can be sorted in priority order (OPT/A3232), and optimal designs can be defined for user-specified cost or performance levels (OPT/A3233). Alternative optimal device designs may then be selected (OPT/A3234) to be compared by the Training Device Selection or Resource Allocation Module to determine which meet the training requirement at the lowest cost. The specific procedure for this activity is described in the detailed description for this node.



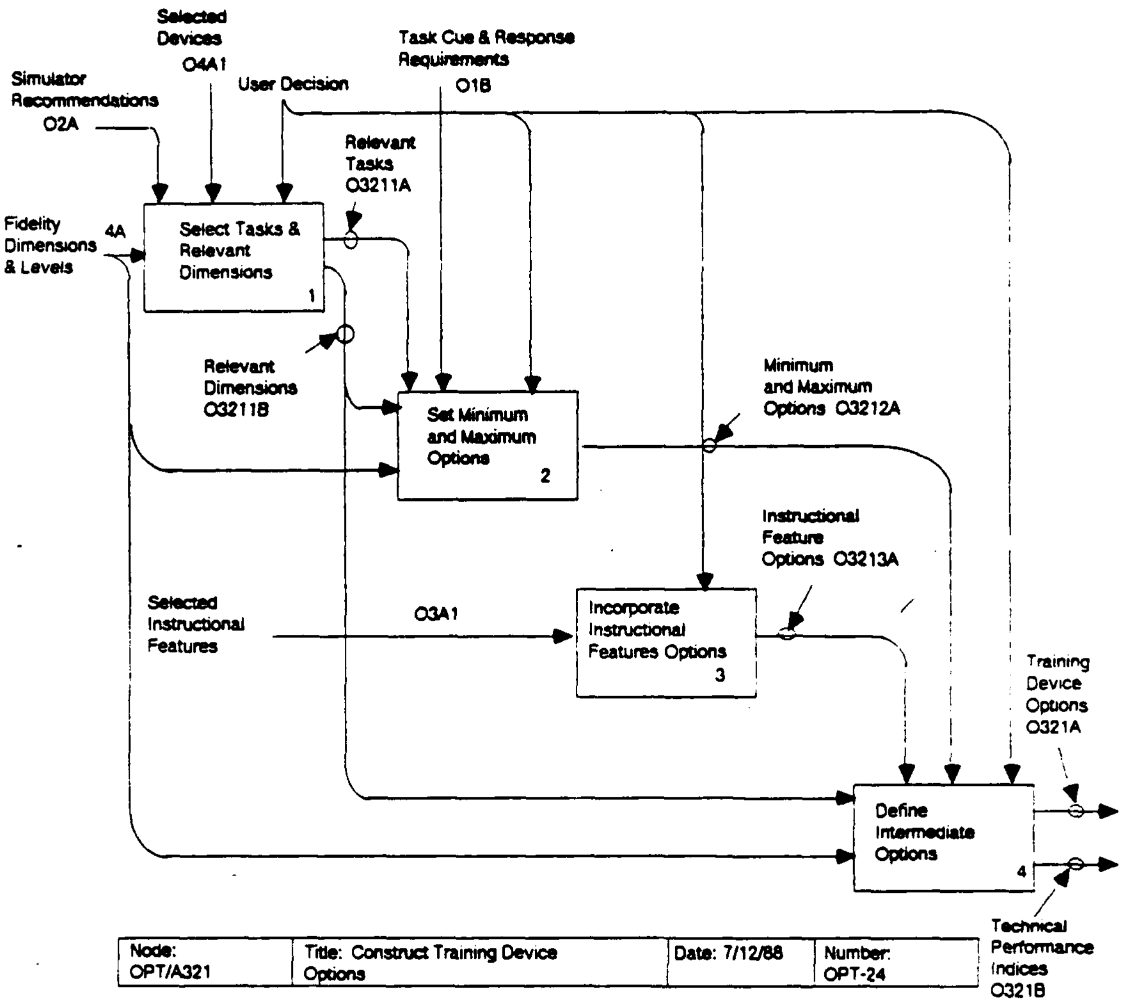
Node: OPT/A32	Title: Optimize Device Fidelity	Date: 7/12/88	Number: OPT-23

OPT/A321: Construct Training Device Options

This activity develops the building blocks that describe options within each of the fidelity dimensions. The user may examine the master list of fidelity dimensions and eliminate those that are inappropriate, set the limits of the options considered by the analysis, and eliminate specific options that are infeasible for reasons outside the scope of the analysis.

OPT/A3211: Select tasks and relevant dimensions. The tasks to be trained on the subject training device are specified by either the Simulation Configuration Module (O2A) or the Training-Device Selection Module (O4A1). The user may modify the outputs of these models, or may select other tasks to be used as the basis for the evaluation. In addition, the user may choose to eliminate some fidelity dimensions (4A) from consideration in the analysis. The output of this activity is a set of relevant tasks (O3211A) and dimensions (O3211B) that will form the basis of the analysis.

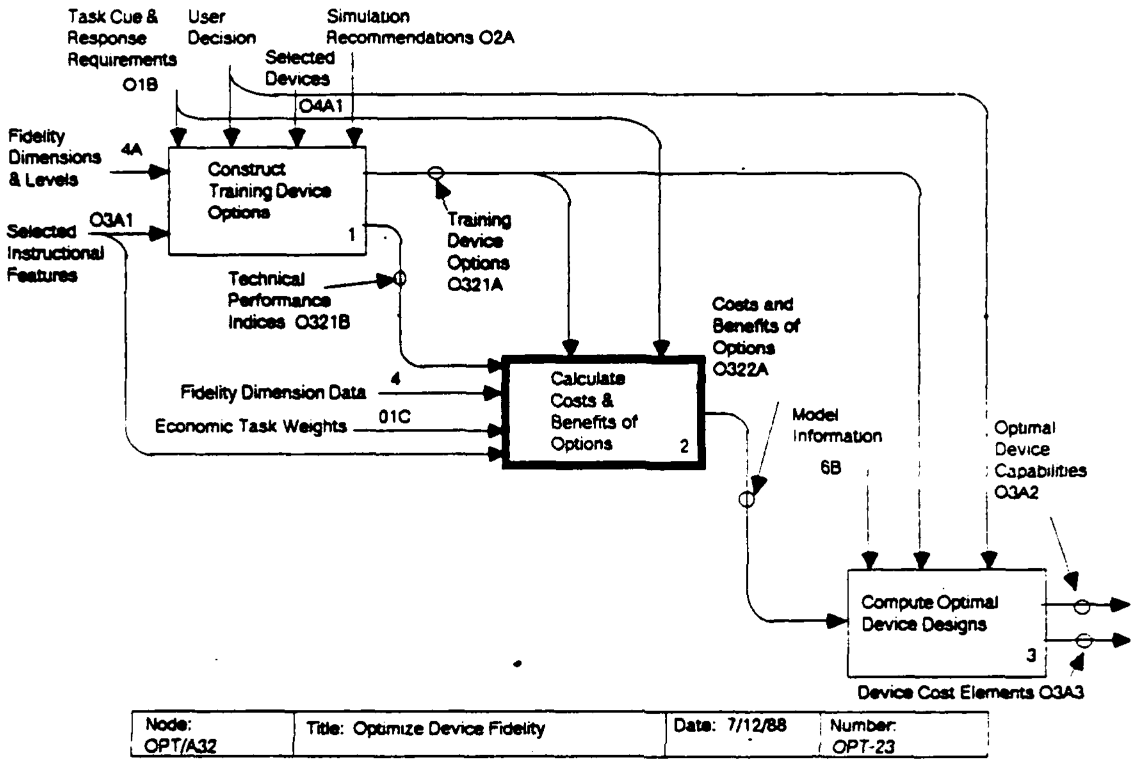
OPT/A3212: Set minimum and maximum options. This process allows the user to compare the options (4A) in each relevant fidelity dimension (O3211B) to the task cue and response requirements (O1B) on each dimension for the selected tasks (O3211A). This comparison should lead the user to either accept



the minimum and maximum options of each dimension contained in the OSBATS data base or specify his own, less-severe minima and/or maxima. The output of this process (O3212A) defines the range of training device designs that will be considered by the analysis.

<u>OPT/A3213: Incorporate instructional-features options</u>. This activity asks the user to decide whether the results of the instructional features model (O3A1), if available, should be incorporated as a fidelity dimension. If the user decides to incorporate instructional features, the costs and benefits of these options are included in the fidelity optimization module. The benefits of these options are computed in OPT/A3224.

<u>OPT/A3214: Define intermediate options</u>. This process allows the user to discard specific intermediate options of any fidelity dimension (4A2). The outputs of this process, which guide and drive the remainder of the analysis, are (1) the set of options to be evaluated in the analysis (O321A) and (2) their associated technical performance indices (O321B; from 4A3).



OPT/A322: Calculate Costs and Benefits of Options

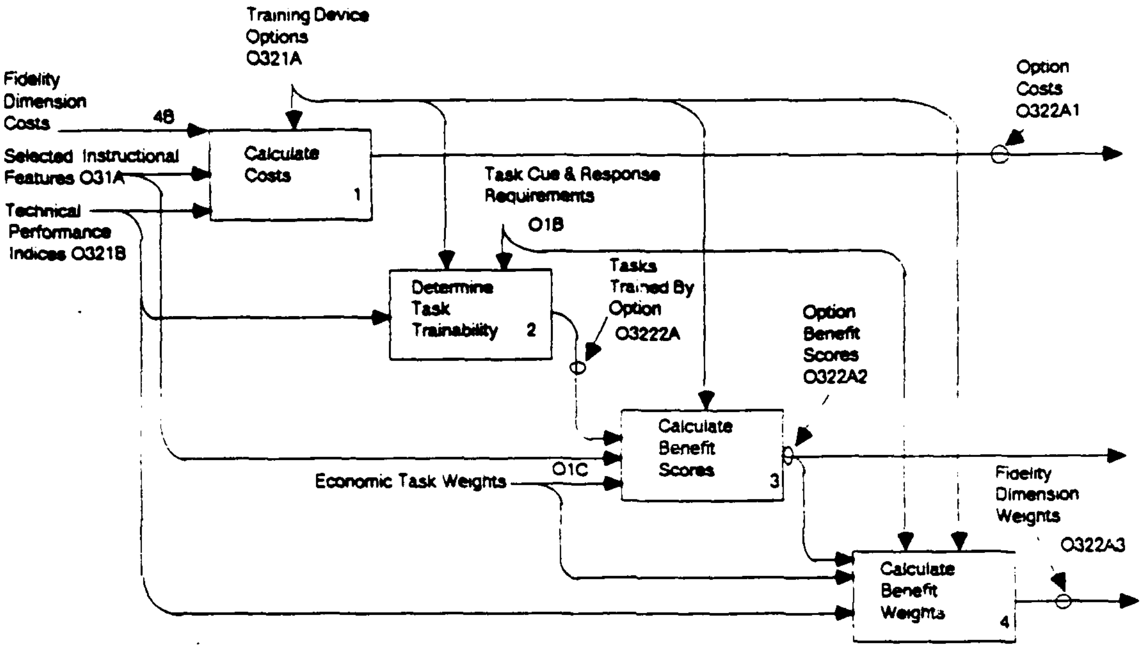
This process yields the benefit and cost data needed to compute the incremental benefit-to-cost ratios for the optimization in OPT/A323.

OPT/A3221: Calculate costs. This activity addresses the development cost of a training device and is a straightforward calculation from fidelity dimension data (4B) using the cost equations shown below. The cost equations relate the development cost to the technical performance index in the range of options under consideration, as follows:

$$CDEV_{jm}(TP_{jm}) = CMIN_j + (CMAX_j - CMIN_j) * CREL_{jm} * TPREL_{jm},$$

where

$CDEV_{jm}$ = the development cost of training device option m on fidelity dimension j,
$CMIN_j$ = the development cost of the lease capable training device option on fidelity dimension j,
$CMAX_j$ = the development cost of the most capable training device option on fidelity dimension j, and
$CREL_{jm}$ = the normalized development cost of training device option m on fidelity dimension j.



Node: OPT/A322	Title: Calculate Costs and Benefits of Options	Date: 7/12/88	Number: OPT-25

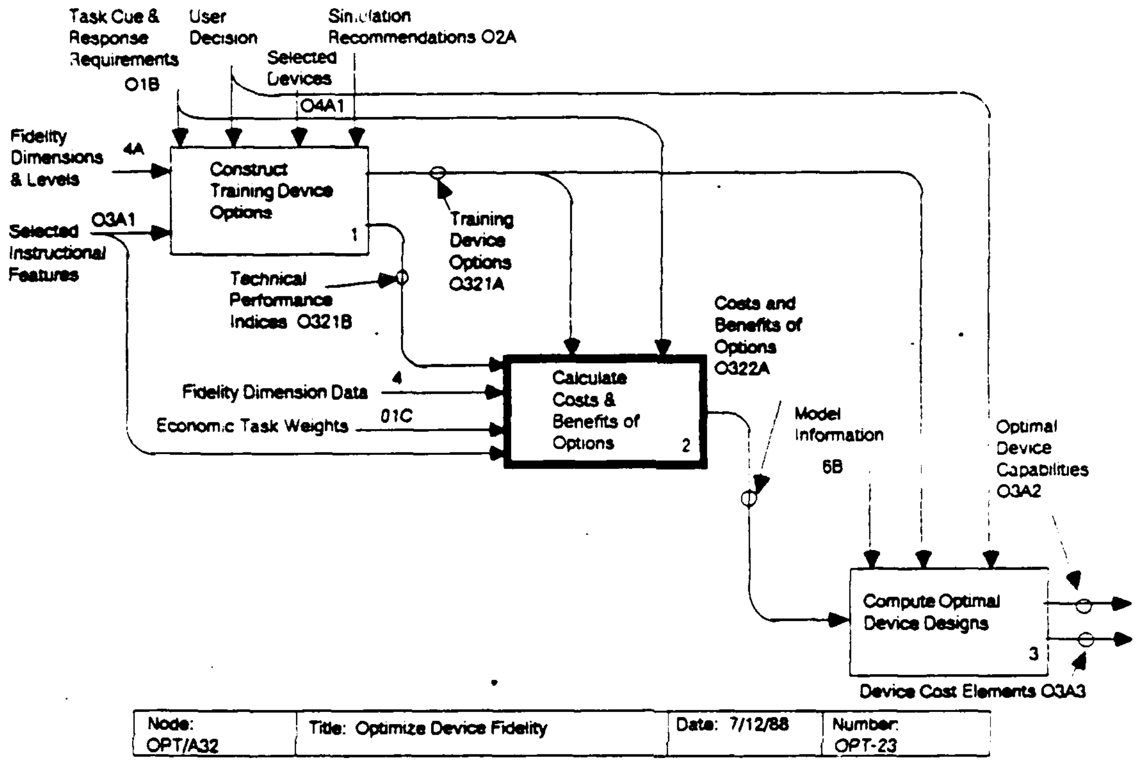
The relative technical performance ($TPREL_{jm}$) is also defined on a scale from 0 to 1.0 as a function of the absolute technical performance (TP), as follows:

$$TPREL_{jm}(TP)_{jm} = \frac{TP_{jm} - TPMIN_j}{TPMAX_j - TPMIN_j}$$

where

TP_{jm} = the technical performance index of training device option m on fidelity dimension j.
$TPMAX_j$ = the technical performance of the most capable option on fidelity dimension j,
$TPMIN_j$ = the technical performance of the least capable option on fidelity dimension j, and
$TPREL_{jm}$ = the normalized (between 0 and 1.0) technical performance of training device option m on fidelity dimension j.

We estimate $TPREL_{jm}$ from the cost using the following equation:



$$TPREL_{jm} = \frac{1 - e^{(-E_j\ CREL_{jm})}}{1 - e^{-E_j}}$$

where

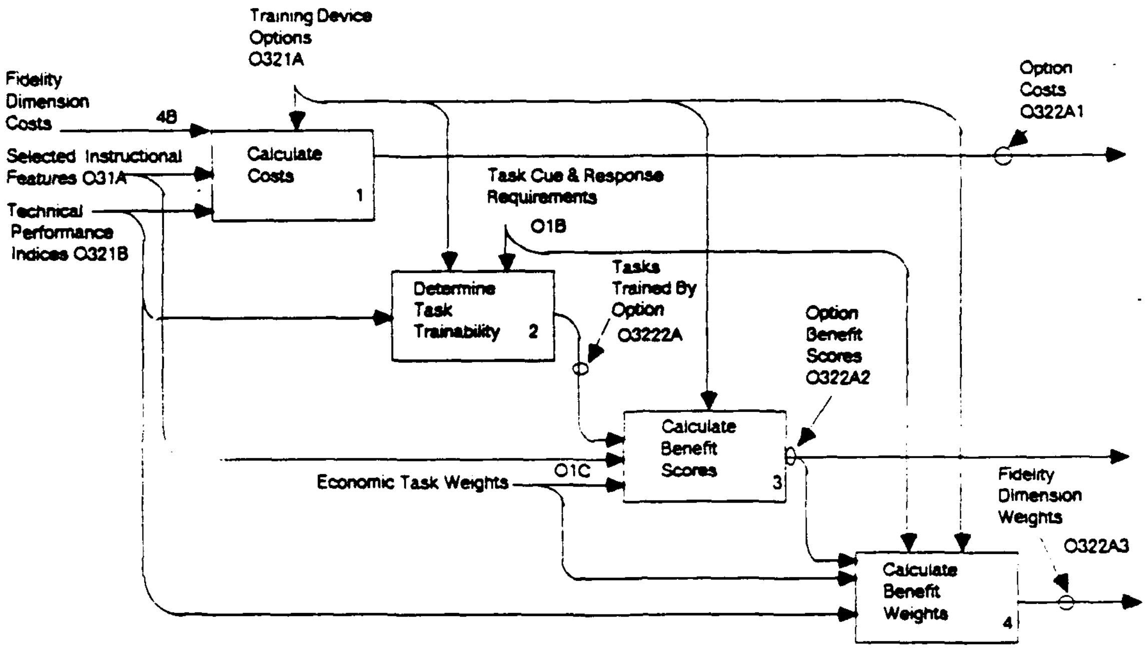
E_j = a constant that describes the fidelity dimension (4B3).

To compute the costs, the inverse of this function would be computed using the following equation:

$$CREL_{jm} = -\frac{1}{E_j}\ln[1 - TPREL_{jm}(1 - e^{-E_j})]$$

This function has the property that technology becomes more costly as the constant exponent (E_j) increases in size.

OPT/A3222: Determine task trainability. This is the first step in the calculation of benefits for training device options. The output variable for this process (TRN_{Tjm}) is the intersection of task T and option m of fidelity dimension j. For each such intersection being considered, the task cue and response requirement (O1B) is compared to the technical performance index (O321B) of the option. If the technical performance index of the



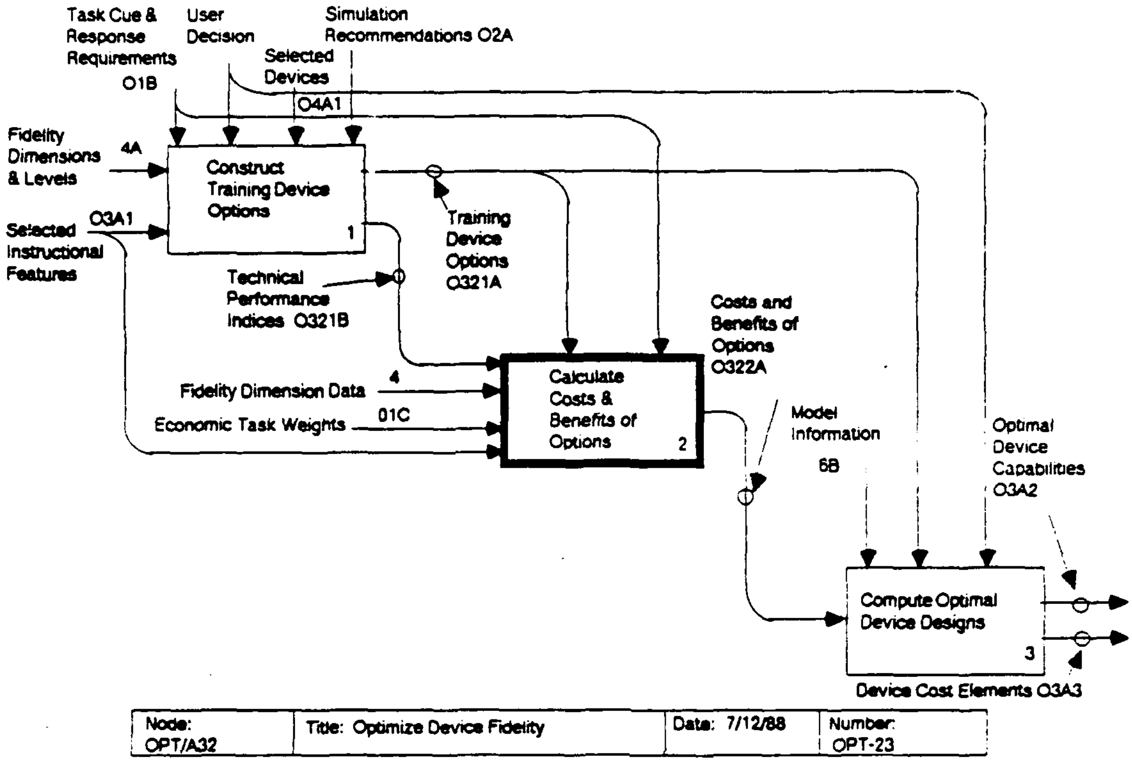
Node: OPT/A322	Title: Calculate Costs and Benefits of Options	Date: 7/12/88	Number: OPT-25

option is greater than or equal to the requirement of the task, TRN_{Tjm} is set equal to 1.0. Otherwise, it is set equal to 0.0. This determination is undertaken for all tasks and all options under consideration.

<u>OPT/A3223: Calculate benefit scores</u>. This process uses the variable TRN_{Tjm} (O3222A) and the task weights (O1C) calculated in OPT/A12 to calculate the benefit score of each option. The benefit score is defined to represent the relative benefit of an option within its fidelity dimension. For all options within a fidelity dimension the benefit scores are normalized to fall between 0 and 1.0; that is, the least effective (or minimum) option is defined to have a benefit of 0 and the most effective (or maximum) option a benefit of 1.0. This normalization defines the benefit to be total benefit above the minimum fidelity option in the dimension and facilitates the development of benefit weights later.

First, the non-normalized benefit scores of the options ($UBEN_{jm}$) are calculated as follows:

$$UBEN_{jm} = \sum_{T} TRN_{Tjm} \, TWT_T$$



where TWT_T is the economic weight of task T. Now the normalized benefit scores of the options ($NBEN_{jm}$) can be calculated as follows:

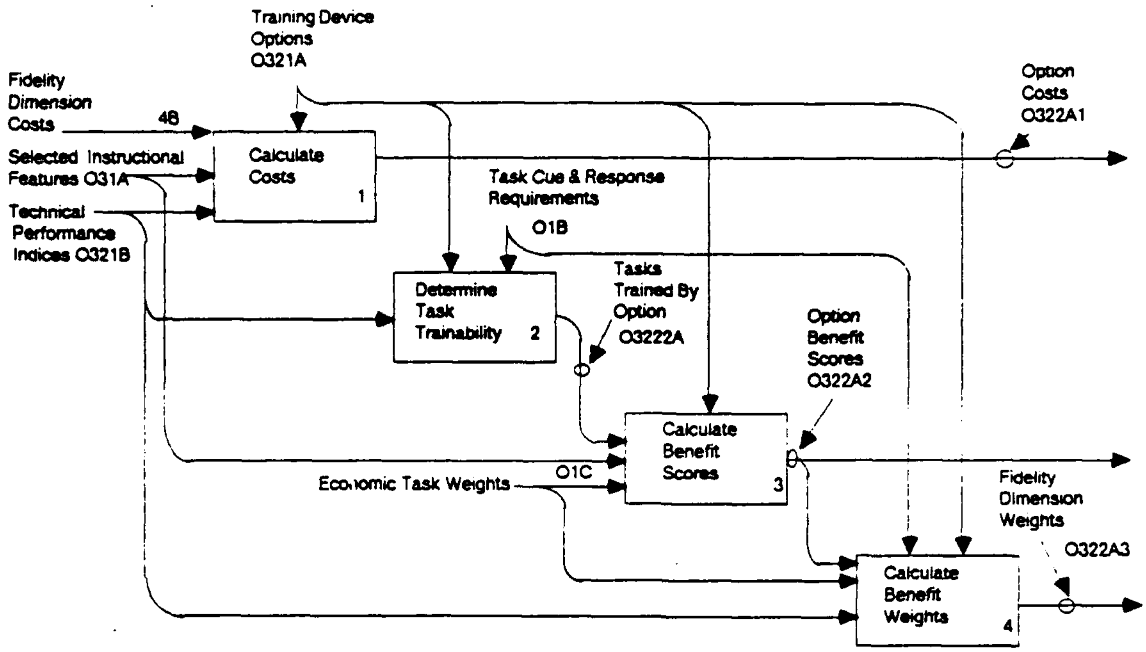
$$NBEN_{jm} = (UBEN_{jm} - UBEN_{j1})/(UBEN_{jM} - UBEN_{j1}),$$

where

$UBEN_{j1}$ = the non-normalized benefit of the minimum option of fidelity dimension j, and
$UBEN_{jM}$ = the non-normalized benefit of the maximum option of fidelity dimension j.

Due to the mathematical operations involved we are assured that $NBEN_{jm}$ does not decrease as m increases for a fixed j. If $NBEN_{jm}$ and $NBEN_{jm+1}$ are equal, then the m+1 option of dimension j should be dropped from consideration.

<u>OPT/A3224: Calculate fidelity benefit weights</u>. This activity uses the task cue and response requirement (O1B) and the task weights (O1C) to calculate a set of fidelity-dimension weights that place benefit on a common scale across fidelity dimensions. First, a modified task cue and response requirement, RM_{ij}, is computed as follows:



Node: OPT/A322	Title: Calculate Costs and Benefits of Options	Date: 7/12/88	Number: OPT-25

$$RM_{Tj} = (FRQT_{Tj} - TP_{j1})/(TP_{jM} - TP_{j1})$$

where

$FRQT_{Tj}$ = is the task cue and response requirement for task T and fidelity dimension j (O1B),

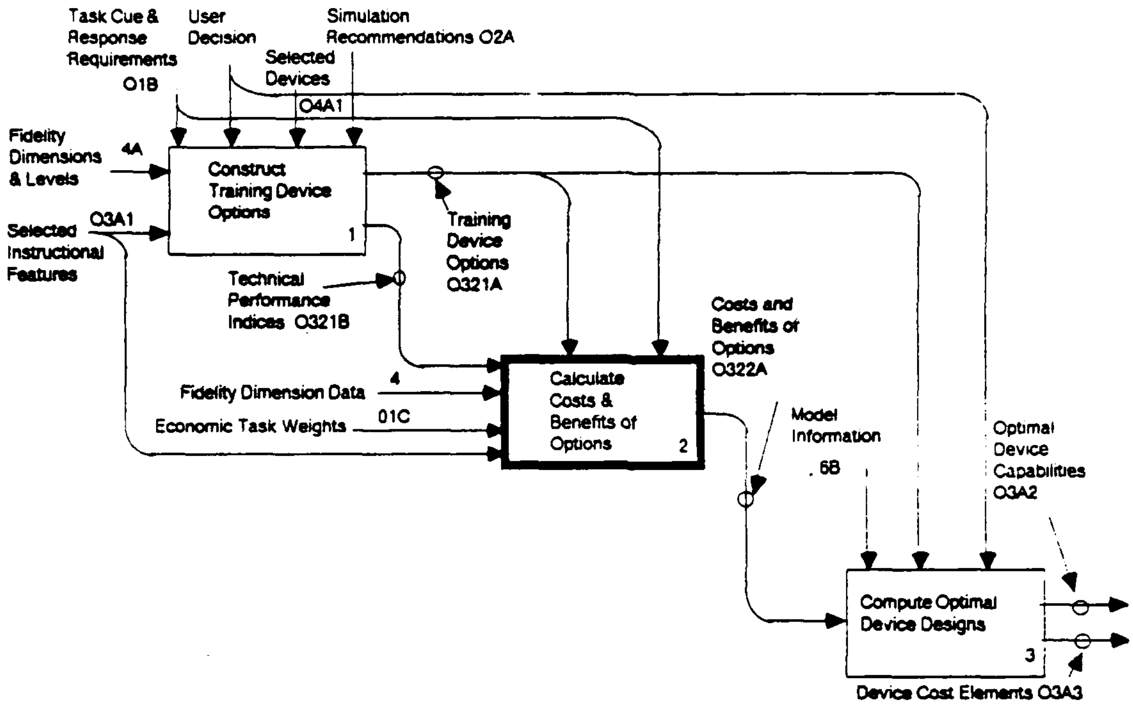
TP_{j1} = the technical performance index of the first level for fidelity dimension j (O321B),

TP_{jM} = the Technical Performance index of the highest level for fidelity dimension j (O321B).

This modification of the task cue and response requirement normalizes the requirement so that it is zero for the least capable option for each fidelity dimension, and one for most capable option. Note that values less than zero should be set to zero. The benefit weights for each fidelity dimension are the sum (across all tasks) of the modified task cue and response requirement times the economic weights. That is,

$$BWT_j = \sum_T RM_{Tj}\ TWT_T$$

These fidelity benefit weights are then normalized to sum to one to maintain scale relationships among benefit values.

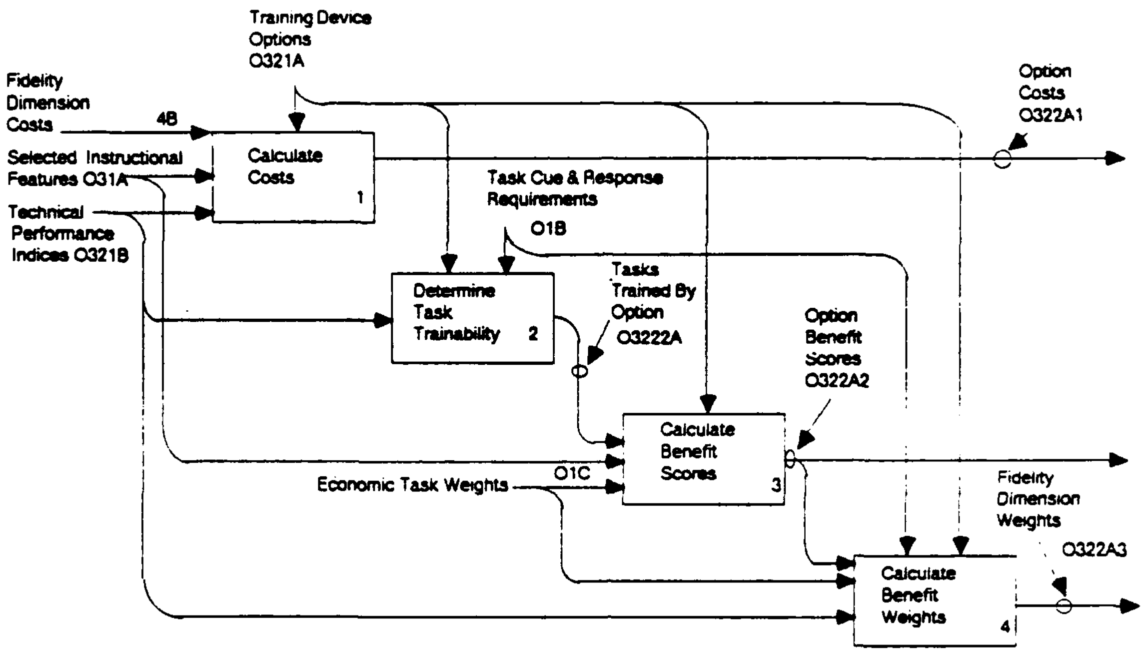


Node: OPT/A32	Title: Optimize Device Fidelity	Date: 7/12/88	Number: OPT-23

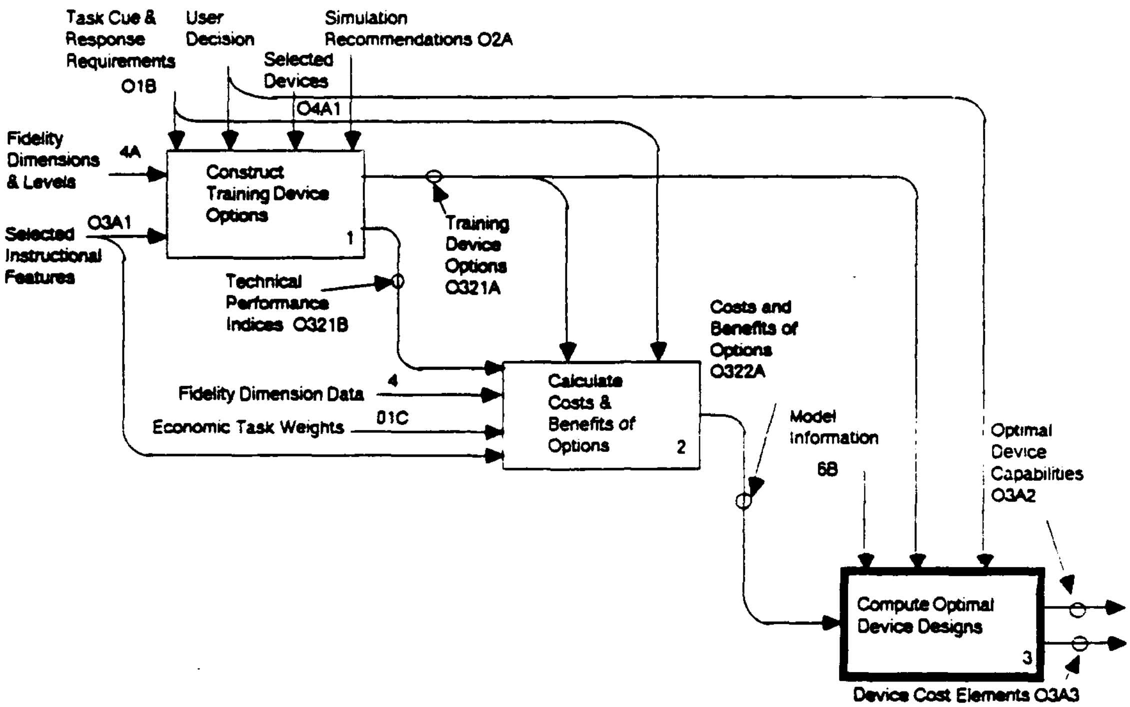
A single value that incorporates the benefit value and the benefit weight for instructional features is calculated based on the relative costs of instructional features compared to the fidelity dimensions. If F_m is defined as the set of instructional features included in option m of the instructional feature dimension, and F_M is the set of instructional features in the maximum option, then:

$$NBEN_{IF,m} = \left[\frac{\sum_{k \varepsilon F_m} BEN_k}{\sum_{k \varepsilon F_M} BEN_k} \right] * \left[\frac{\sum_{k \varepsilon F_M} IFC_k}{\sum_{j} CMAX_j} \right] * 100$$

The first term computes the fraction of the benefits associated with the instructional features (BEN_k) included in option m compared to the maximum option, M. The second term computes the ratio of the maximum cost of the instructional features being considered (IFC_k) to all of the costs associated with the fidelity dimensions ($CMAX_j$). The 100 is used to place all of the weighted scores on the same scale. Thus, no further weight is applied to the instructional feature variable ($BWT_{IF} = 1$).



Node: OPT/A322	Title: Calculate Costs and Benefits of Options	Date: 7/12/88	Number: OPT-25



Node: OPT/A32	Title: Optimize Device Fidelity	Date: 7/12/88	Number: OPT-23

OPTA323: Select Optimal Device Designs

This process assembles the device fidelity-dimension levels based upon the incremental benefit-to-cost ratios to attain these levels. It uses this information to determine the optimal training device design at any user-specified level of development cost or benefit. These designs may be evaluated further by using them as inputs to the Training-Device Selection Module or Resource Allocation Module, to determine which option meets the training requirements at the least cost.

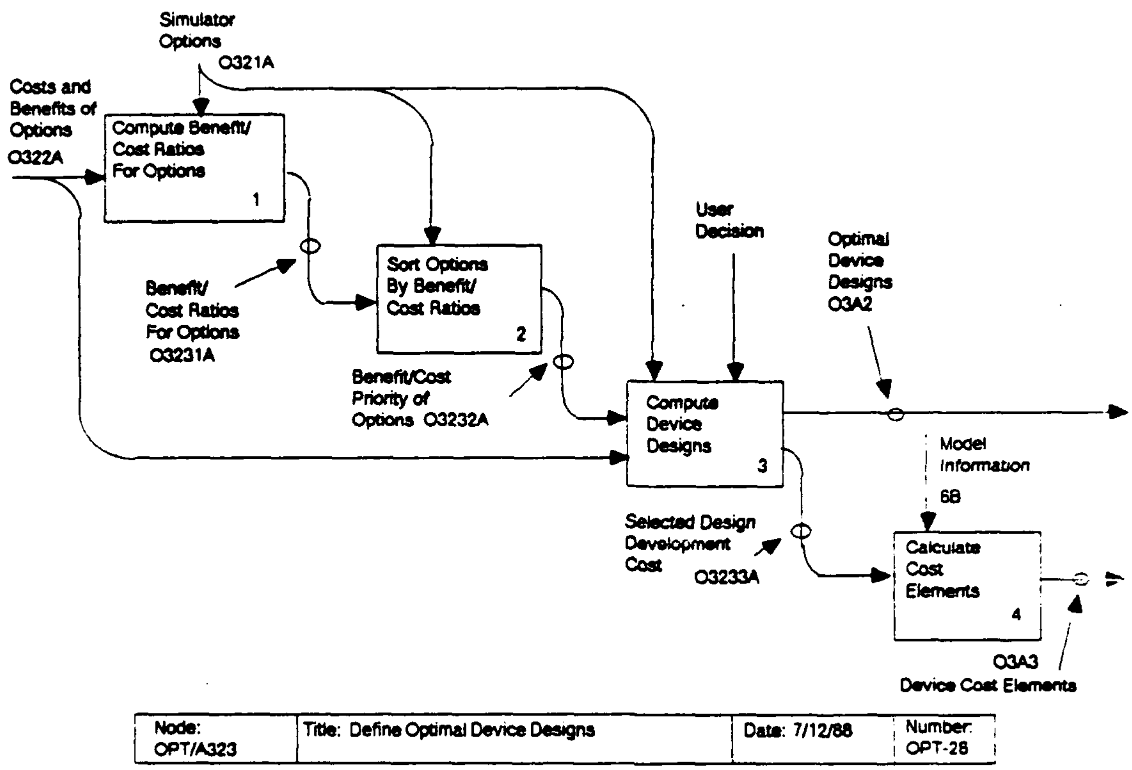
OPT/A3231: Compute benefit-to-cost ratios for the options. This activity uses the cost and benefit data from the previous major process (O322A). The weighted incremental benefit-to-cost ratio of option m from fidelity dimension j, $(B/C)_{jm}$, is defined as follows:

$$(B/C)_{jm} = BWT_j * (NBEN_{jm} - NBEN_{jm-1})/(CDEV_{jm} - CDEV_{jm-1}),$$

where

$CDEV_{jm}$ = the development cost of option m in dimension j.

Before the options can be sorted by their benefit-to-cost ratios, these ratios must be guaranteed to be decreasing within each



fidelity dimension. That is, $(B/C)_{jm}$ must be greater than $(B/C)_{jm+1}$. This is done by dropping any level m from consideration that does not meet this test of decreasing benefit/cost ratio and setting $(B/C)_{mk+1}$ equal to

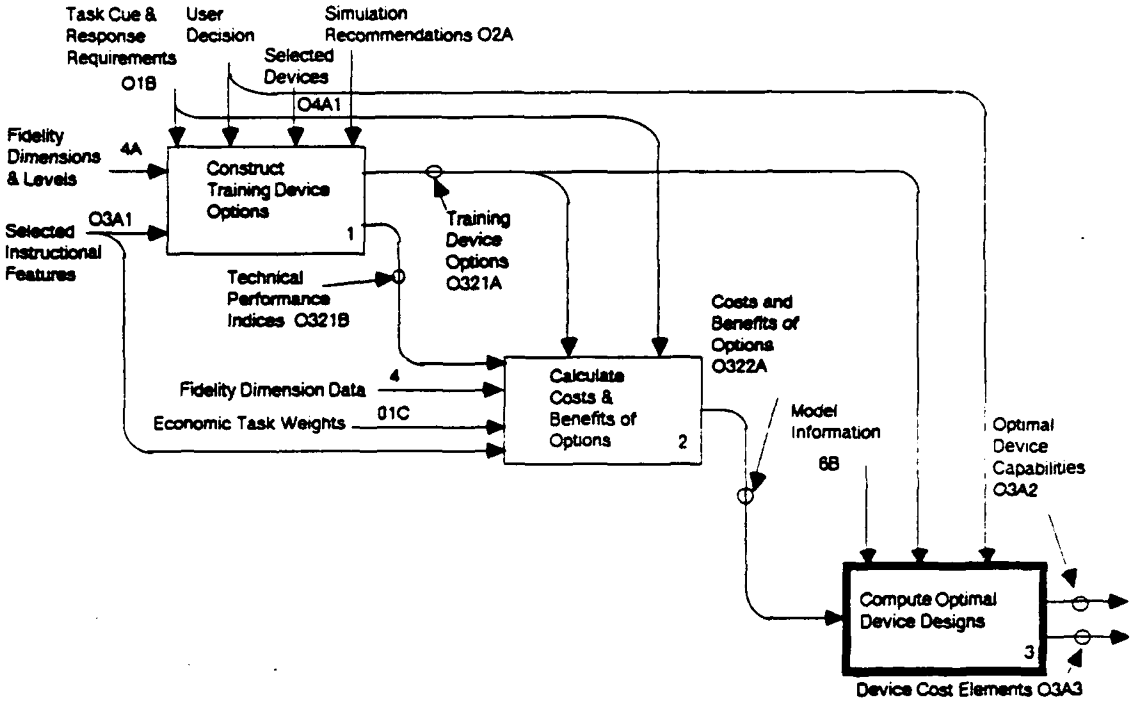
$$BWT_j * (NBEN_{jm+1} - NBEN_{jm-1})/CDEV_{jm+1} - CDEV_{jm-1}).$$

This recomputation must continue until all levels within a fidelity dimension that have not been dropped satisfy this test.

<u>OPT/A3232: Sort options by benefit-to-cost ratios</u>. Because these options have been constructed to be independent building blocks of a training device (to the extent that this is possible), they can be prioritized solely on the basis of the incremental $(B/C)_{jm}$. The output of this activity (O3232A) lists the options on all fidelity dimensions in order of decreasing $(B/C)_{jm}$.

<u>OPT/A3233: Compute device designs</u>. This process takes input from the user as to the target cost or benefit levels of device designs that need to be considered. Associated with the prioritized list of options will also be the appropriate weighted incremental benefit,

$$(BWT_j * (NBEN_{jm} - NBEN_{jm-1})),$$



Node: OPT/A32	Title: Optimize Device Fidelity	Date: 7/12/88	Number: OPT-23

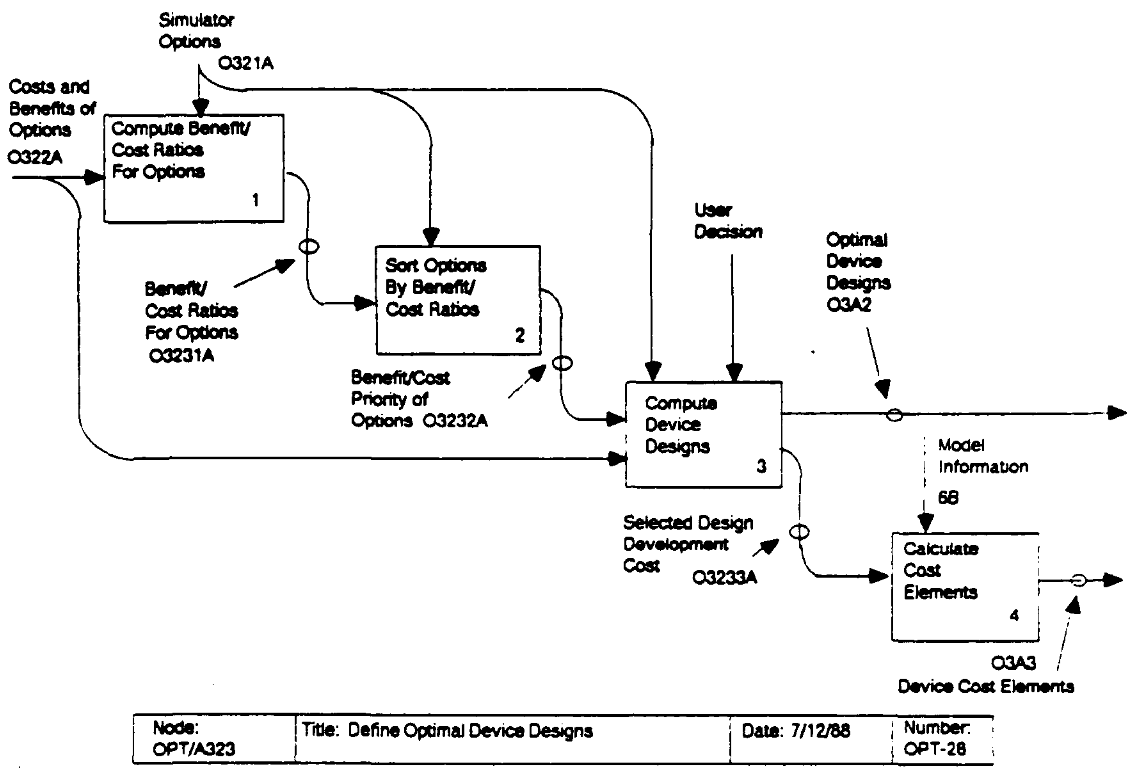
and incremental cost,

$(CDEV_{jm} - CDEV_{jm-1})$,

of each option. These incremental costs and benefits can be accumulated down the priority list to determine the total cost and benefit of a device design that contains the given option and all higher priority options above it. At any time the user can re-enter this model and request new device designs by giving new cost or benefit levels. The user may select one or more of the optimal device designs to accept or to evaluate further using the Training Device Selection Module or Resource Allocation Module.

OPT/A3234: Calculate cost elements. The costs of the optimal training devices selected by the user represent development cost only. In this activity, the development costs are accumulated, and the other cost elements are estimated. The total investment cost (INV_i) for the device is simply the sum of the $CDEV_{jm}$ for the selected levels of each fidelity dimension. The annual fixed operating cost (FOC_i) and hourly variable operating cost ($VHRC_i$) are derived from INV_i, as follows:

$FOC_i = INV_i / FCFAC$

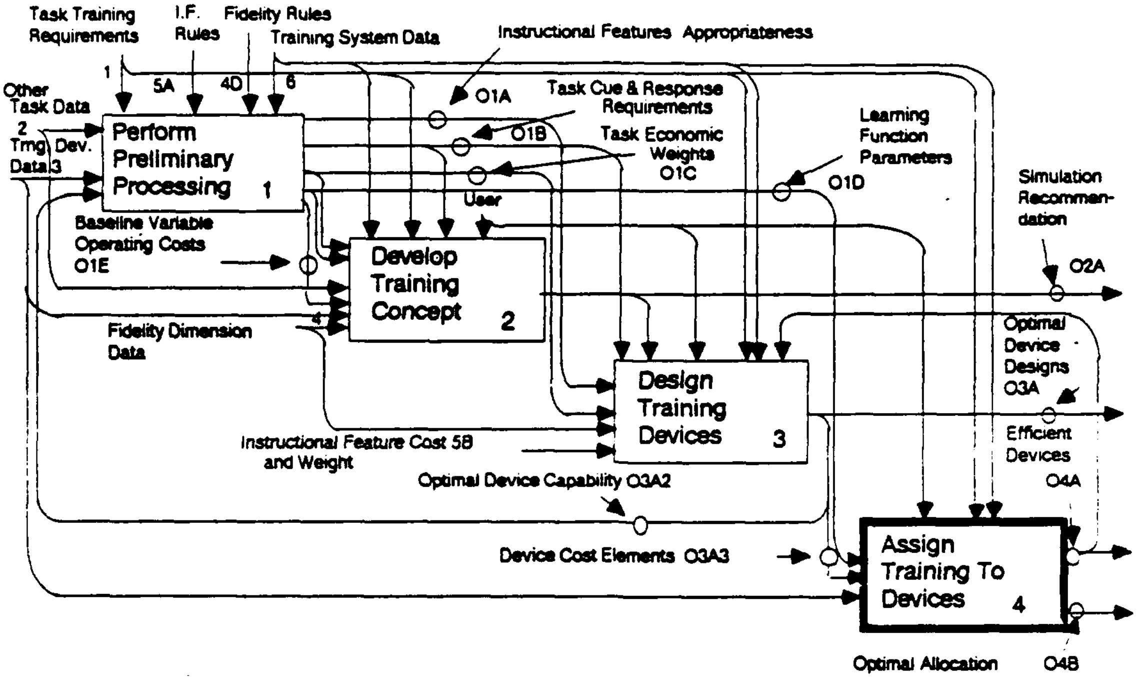


$VHRC_i = INV_i \; / \; VCFAC$

where

FCFAC = a constant fixed-cost estimation factor (6B8), and
VCFAC = a constant variable-cost estimation factor (6B9).

Device utilization ($UMAX_i$) and life cycle (LC_i) are set to the constant values, UEST (6B10) and LCEST (6B11), respectively.

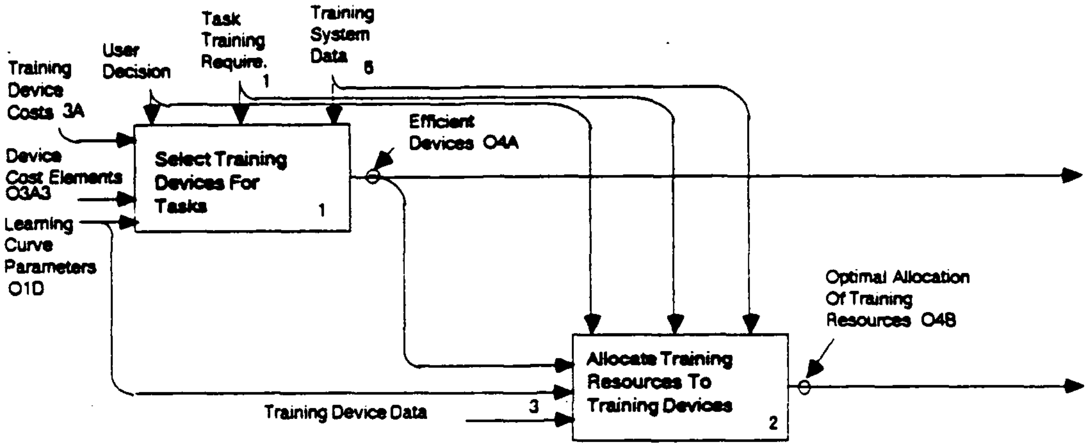


Node: OPT/A0	Title: Optimize Training System Design	Date: 7/27/88	Number: OPT-2

OPT/A4: Assign Training to Devices

The goal of this activity is to assign training on each task among previously defined training devices. These training devices may have been designed by the modules in OPT/A3, or they may be templates of existing devices. In either case, the activity must determine the allocation scheme that meets the training requirements at the minimum cost. Two tools were designed to address this problem. The first tool makes simplifying assumptions to allow interactive operation. For example, this tool assumes that as many devices as needed may be acquired to meet training time demands. The second tool makes fewer assumptions, and allows the user to impose additional constraints on training devices usage. For example, it considers the discrete costs of device acquisition, and it allows the user to place a ceiling on the number of devices of each particular type that may be acquired.

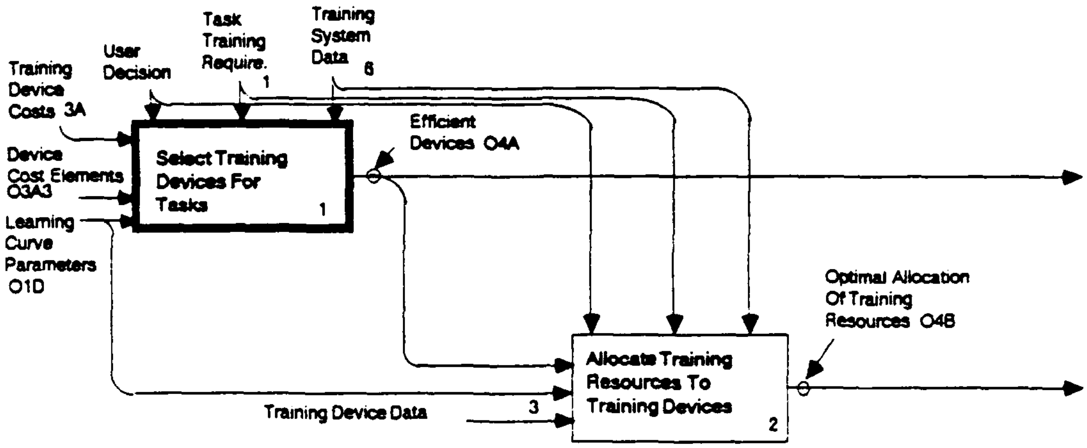
OPT/A41: Select training devices for tasks. This tool, the Training Device Selection Module, determines a group of devices that should be used to train each task, and calculates the amount of time a student should spend training on each of these devices. The chosen training devices are conditionally optimal, in that they meet the training requirements at the minimum cost subject to the simplifying assumptions mentioned above. Those



Node: OPT/A4	Title: Assign Training To Devices	Date: 7/12/88	Number: OPT-27

assumptions imply that training device cost functions are simple linear functions of per-student training device use (time). This simplification allows the module to perform the optimization in seconds.

OPT/A42: Allocate training resources to training devices. This tool, the Resource Allocation Module, also determines device usage to minimize total training costs, but treats the cost relations in more detail and allows the imposition of certain constraints. In particular, it weakens the assumption of simple linear cost functions and allows for the inclusion of assets on hand. The new form for a training device cost function is assumed to be piecewise linear, with (possibly) an initial segment corresponding to assets on hand, and the subsequent segments corresponding to acquisition of another one of that device. The optimization conducted in this module considers constraints that limit the amount of time a training device may be used or the performance level at which the device may be used. As a result of the increased generality, the Resource Allocation Module will require significantly greater time for solution than the Training Device Selection Module. Consequently, its use may be more appropriate in a "batch" mode than in an interactive mode. Provision is made, however, to speed the algorithm by sacrificing some of the detailing in cost function representation.

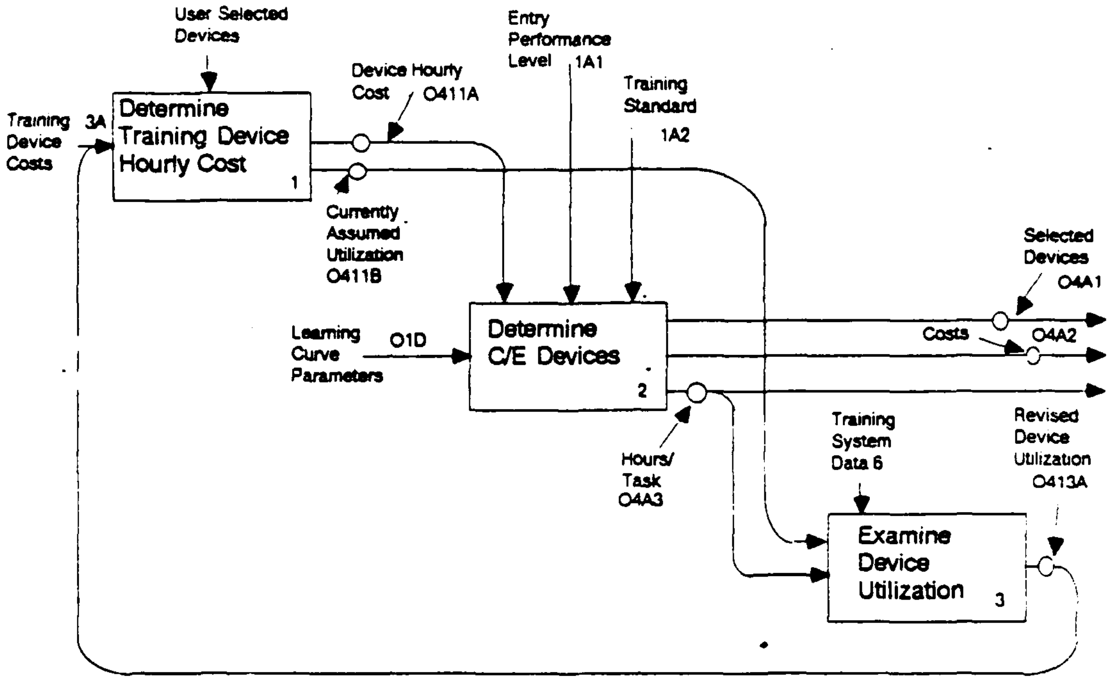


Node: OPT/A4	Title: Assign Training To Devices	Date: 7/12/88	Number: OPT-27

<u>OPT/A41: Select Training Devices for Tasks</u>

This tool applies cost-effectiveness analysis to select the training devices that meet training requirements at the minimum cost. The model considers (1) the fidelity of each candidate training device, (2) the instructional features present in each candidate training device that can reduce the time required to reach a training criterion, and (3) the level of training to be conducted on each training device. The model determines the group of devices that should be used to train each task, and the level of training at which a student should change from one device to the next device in sequence, in order to achieve required training standards at minimum total training cost. Model outputs include the associated training time required on each device.

<u>OPT/A411: Determine training device hourly cost</u>. This activity allocates the total life cycle cost of a training device over its expected number of hours use. The estimate is based on a nominal utilization ($UNOM_i$) in hours per year. In the initial run of the model $UNOM_i$ is set to the maximum annual utilization ($UMAX_i$) found in the data base (3A5) or calculated in OPT/A3234 (O3A3). On later iterations, the value of $UNOM_i$ that is updated in activity OPT/A4134 (O413A) is used in this activity. The



Node: OPT/A41	Title: Select Training Device For Tasks	Date: 8/1/88	Number: OPT-28

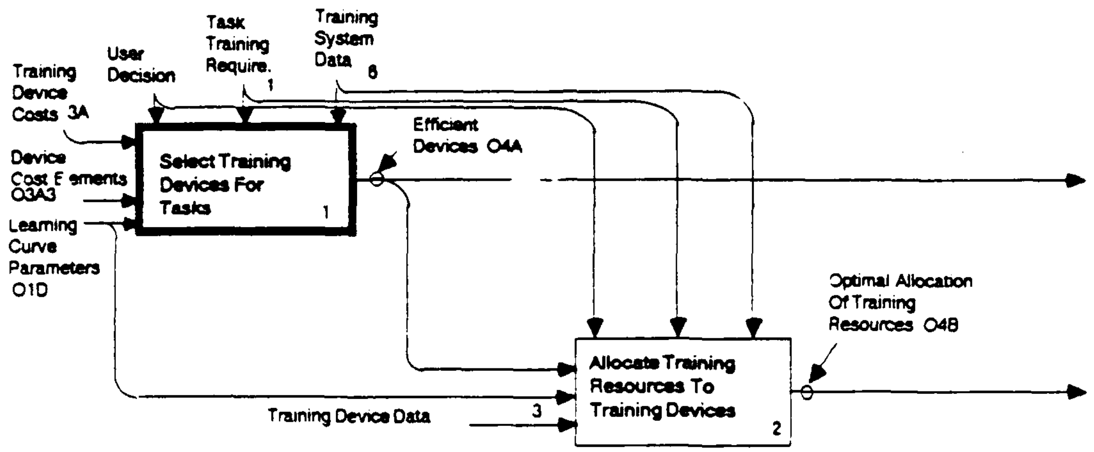
total hourly cost for the device ($VTOTC_i$) is given by the following equation:

$$VTOTC_i = INV_i/(LC_i * UNOM_i) + FOC_i/UNOM_i + VHRC_i,$$

where the following parameters are taken from the data base:

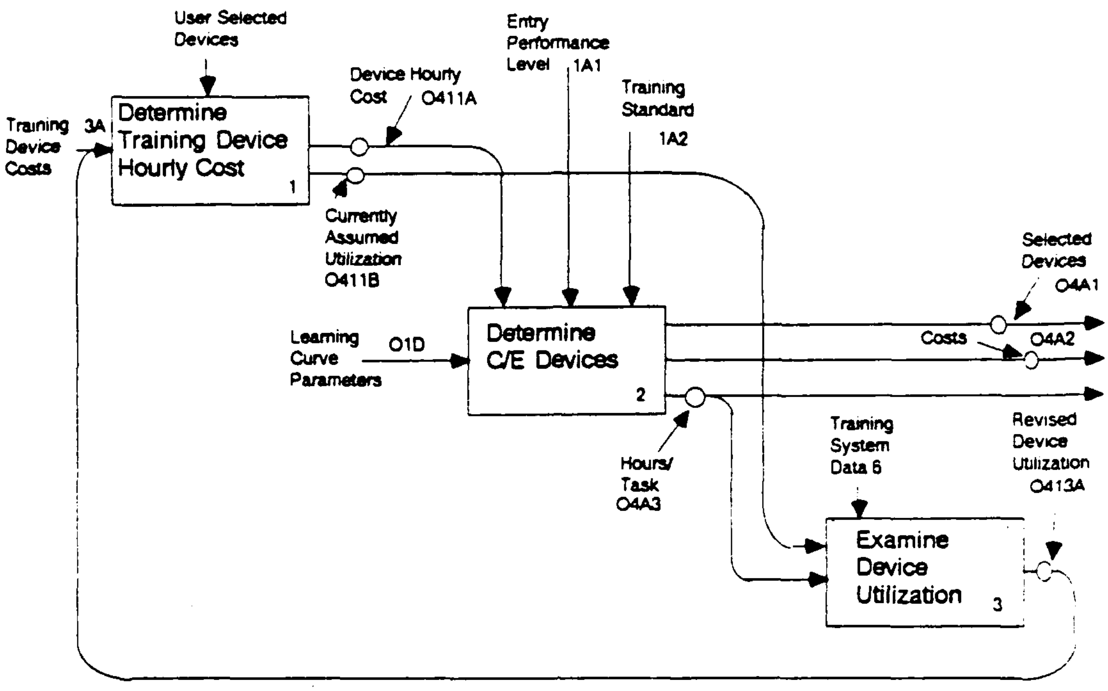
INV_i = the investment cost per device for device i (3A1),
LC_i = the number of years in the life cycle for device i (3A4),
FOC_i = the annual fixed operating cost of device i (3A2), and
$VHRC_i$ = the hourly variable operating cost of device i (3A3).

OPT/A412: Identify cost effective devices. This activity analyzes the relationship between each task and possible training device configurations (selection and sequence) to determine the training device configuration that can train from entry level to training standard at the lowest cost. It performs this analysis by choosing, for each performance level, that device providing the highest rate of increase in student performance as a function of the effective price of using that device. The specific procedure used is described in the detailed description of this activity.

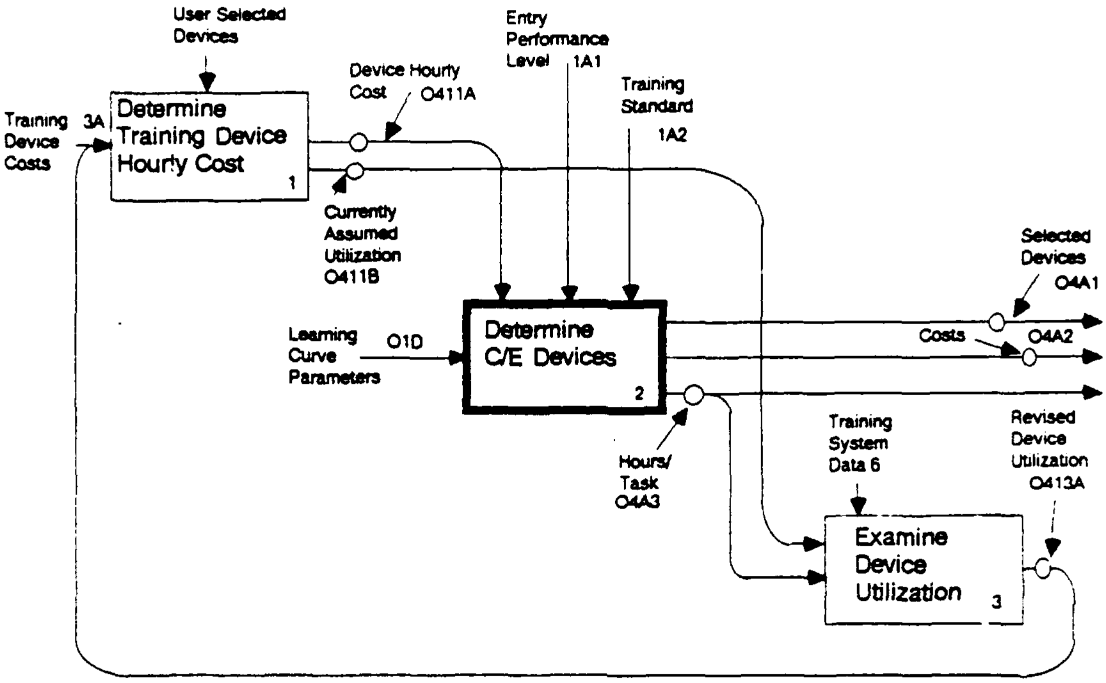


Node: OPT/A4	Title: Assign Training To Devices	Date: 7/12/88	Number: OPT-27

<u>OPT/A413: Examine Device Utilization</u>. This activity compares the value of device utilization that was assumed in the analysis ($UNOM_i$) with an actual utilization value, $UACT_i$, calculated at OPT/A4133. If $UNOM_i$ and $UACT_i$ are very different for device i, then that device may be considerably more or less efficient than was assumed by the analysis. In this case the user may decide to rerun the analysis adjusting $UNOM_i$ to have the value of $UACT_i$. The specific procedure is described in the detailed diagram for this node.



Node: OPT/A41	Title: Select Training Device For Tasks	Date: 8/1/88	Number: OPT-28



Node: OPT/A41	Title: Select Training Device For Tasks	Date: 8/1/88	Number: OPT-28

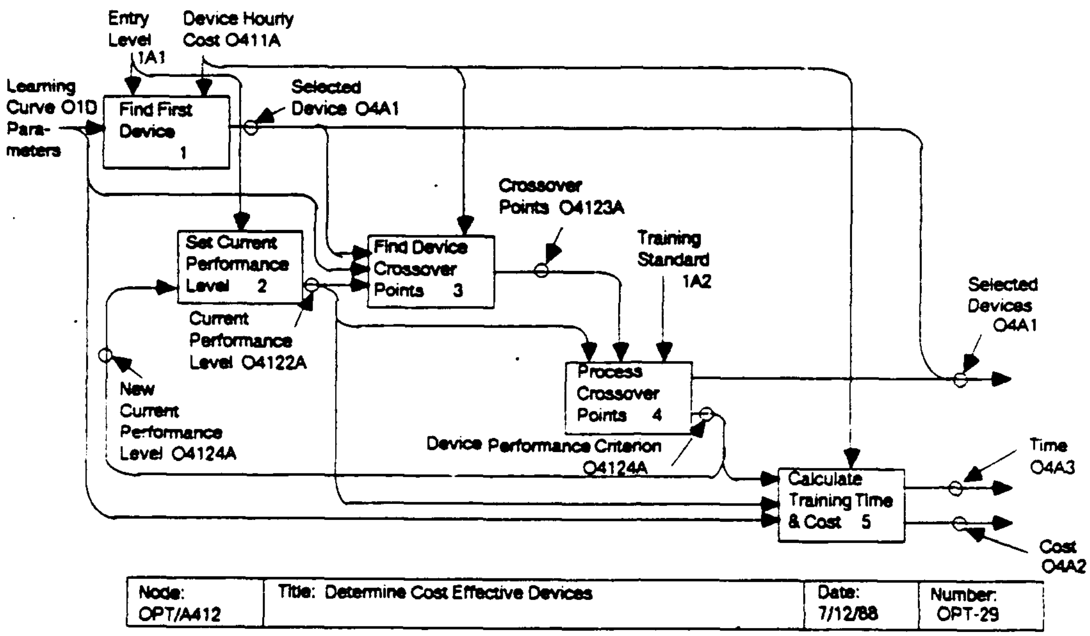
OPT/A412: Identify Cost Effective Devices

This activity identifies one or more efficient training devices for each task. The basic method used maximizes the slope of the curve that expresses performance on a task as a function of cost for each training device. At any performance level, the method chooses the training device with the highest slope, and continues with that device until the slope for that device is exceeded by another device, or until the training standard is reached. The cost effective devices are determined using the following five steps.

OPT/A4121: Determine first device. The first device that is chosen maximizes the slope of the learning curve with respect to cost at the entry performance level. The slope of the learning curve with respect to attributed per-student cost, C, expressed as a function of student performance level, P, is given by the following equation:

$$\frac{dP}{dC} = \frac{ASM_{Ti} * r * TM_{Ti} * TSF_{T}}{VTOTC_{i}} (1 - P/ASM_{Ti})^{(r + 1)/r}$$

where VTOTC is the hourly cost rate, found at OPT/A411, and where the following learning curve parameters used (O1D):

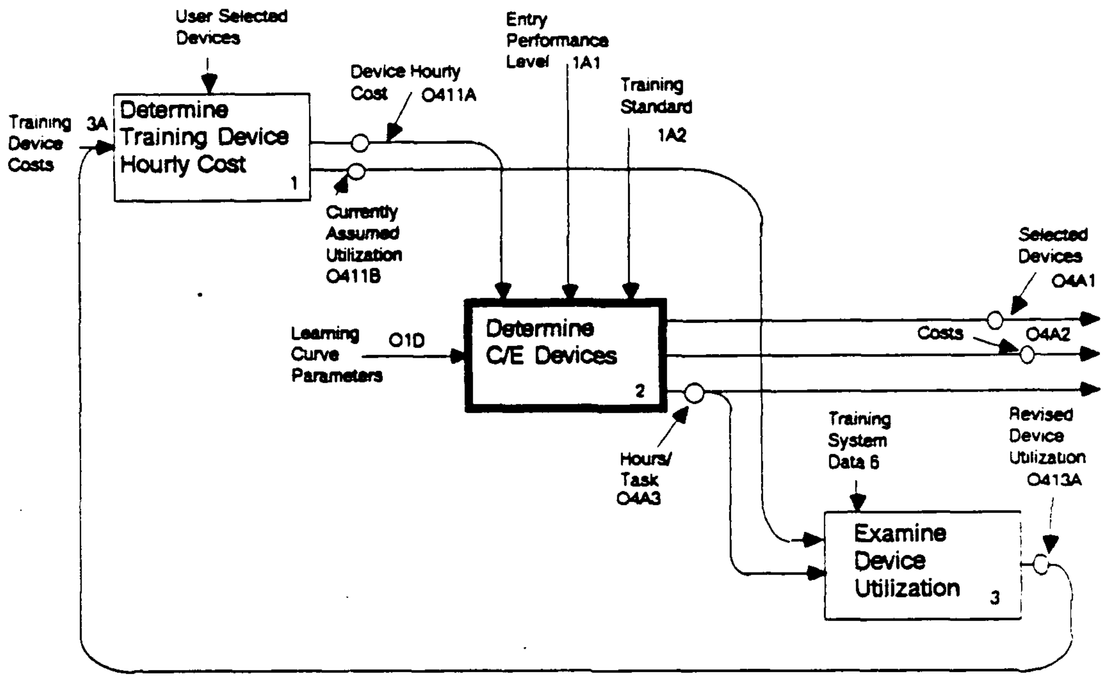


ASM_{Ti} = asymptote of learning curve for task T and device i,
r = learning curve exponent,
TM_{Ti} = learning curve time multiplier for task T and device i,
TSF_T = time scaling factor.

Solving this equation at $P = ENT_T$ gives the cost-effectiveness of each training device at the entry level. The activity then chooses the device with the highest slope. The currently chosen device will be termed the incumbent device and will be denoted by the subscript, s.

<u>OPT/A4122: Set current performance level</u>. The current performance level (P_{now}) keeps track of the effect of training on performance. Initially, the current performance level is set to the entry performance level (ENT_T). As training is conducted on a training device, performance improves. When a change is made from one training device to another training device (in OPT/A4124), P_{now} is set to the performance level at the slope crossover point (determined in OPT/A4123).

<u>OPT/A4123: Find device crossover points</u>. This activity determines the performance level at which the slope of the learning curve for each device surpasses that of the curve for



Node: OPT/A41	Title: Select Training Device For Tasks	Date: 8/1/88	Number: OPT-28

the incumbent device, s. The crossover point (PX_{si}) is defined by the following equation:

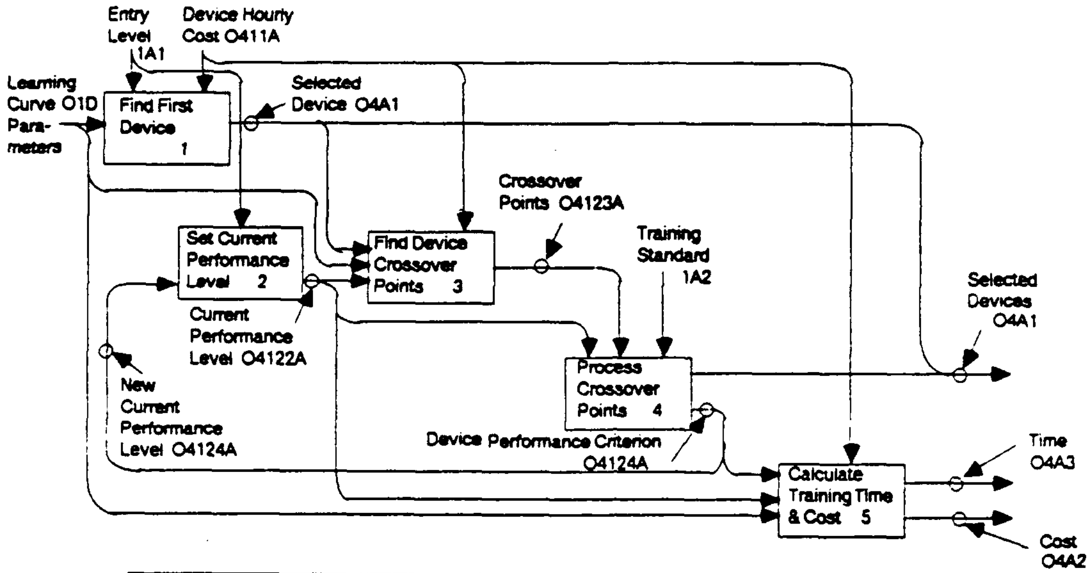
$$PX_{si} = \frac{ASM_{Ti} - (ASM_{Ts} * z)}{1 - z},$$

where

$$z = \frac{ASM_{Ti}}{ASM_{Ts}} \left[\frac{ASM_{Ts}}{ASM_{Ti}} \frac{TM_{Ts}}{TM_{Ti}} \frac{VTOTC_i}{VTOTC_s} \right] r/(r + 1)$$

<u>OPT/A4124: Process crossover points</u>. This activity examines the crossover points, eliminates training devices that are no longer in contention for this task, and selects the best candidate training device for further training. The activity uses the following rules to process the crossover points:

1. If $PX_{si} < P_{now}$, then there will be no switch to training device i for greater performance levels, so eliminate it from further consideration.



Node: OPT/A412	Title: Determine Cost Effective Devices	Date: 7/12/88	Number: OPT-29

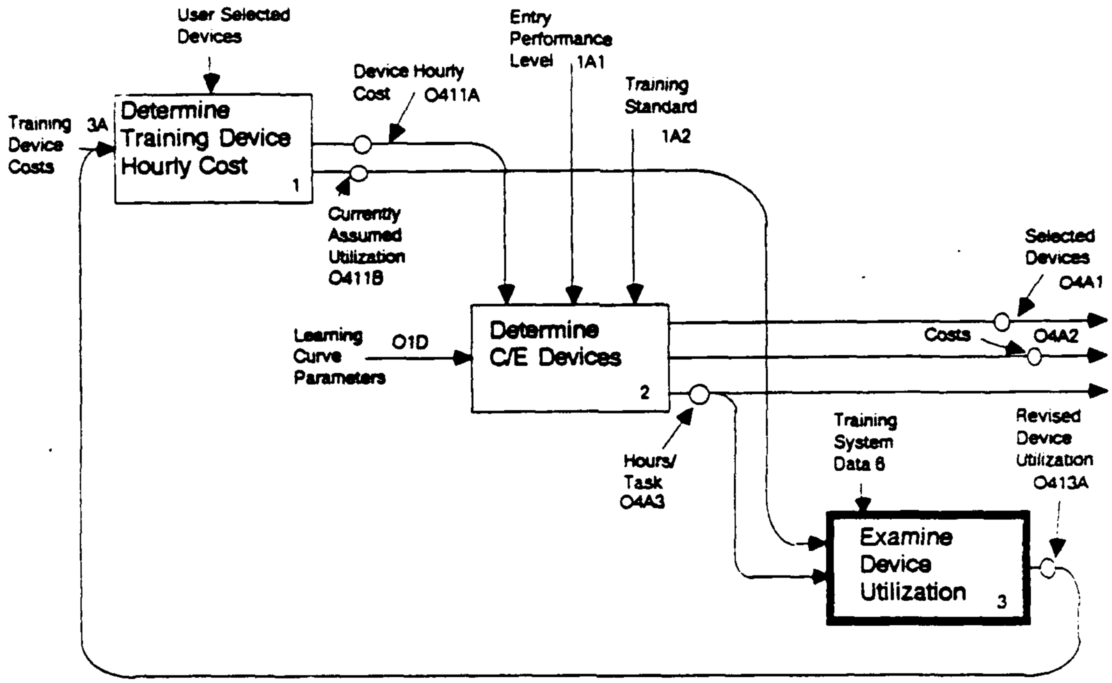
2. If no candidate training device survives the previous step, then train on the incumbent device to the training standard. If the asymptote of the incumbent device is less than STD_T, then the performance standard cannot be met using the combination of devices being evaluated.

3. Otherwise, choose as the new incumbent device the device whose crossover point is reached first, i.e., that device with the smallest PX_{si}. Set the training criterion for device s to this minimum PX_{si}. This value will also become the new current performance level when it is processed by OPT/A4122, but only after completion of OPT/A4125.

<u>OPT/A4125: Calculate training time and cost</u>. This activity determines the time required to train from P_{now} to the next crossover performance level (i.e., the minimum PX_{si}) on the incumbent device, s. The training time ($TRNT_{Ti}$) is calculated from the inverse of the learning function, as follows:

$$TRNT_{Ts} = \frac{1}{TM_{Ts}\ TSF_T}\left[(1 - PX_{si}/ASM_{Ts})^{-1/r} - (1 - P_{now}/ASM_{Ts})^{-1/r}\right]$$

The cost is calculated by multiplying $TRNT_{Ts}$ by the total hourly training device cost, $VTOTC_s$.



Node: OPT/A41	Title: Select Training Device For Tasks	Date: 8/1/88	Number: OPT-28

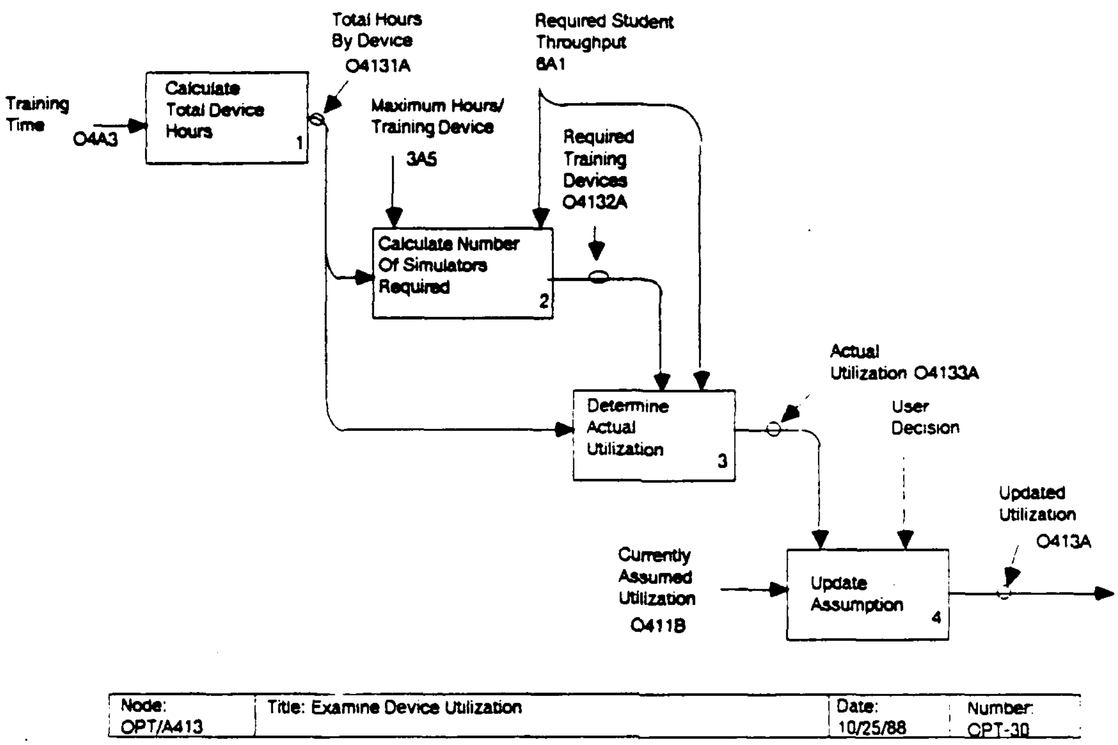
OPT/A413: Examine Device Utilization

In this activity the solutions for the individual tasks are aggregated, and an overall accounting of the devices, costs, and training time is produced. In addition, assumptions of the model regarding training simulator utilization are checked, and the actual utilization is presented to the user, so that the user may decide whether reanalysis is required.

OPT/A4131: Calculate total device hours. This step accumulates the total training time ($TRNT_{Ti}$) over tasks. The resulting device training time ($DEVT_i$) is simply the sum of $TRNT_{Ti}$ over all tasks.

OPT/A4132: Calculate required number of training devices. The first step in calculating the training device utilization ($UACT_i$) is to determine the number of training devices required to implement the selection ($NDEV_i$). $NDEV_i$ is calculated from the device training time ($DEVT_i$) and the maximum device utilization ($UMAX_i$) using the following equation.

$$NDEV_i = \text{ceiling}\left(\frac{DEVT_i * GRADS}{UMAX_i}\right)$$



where

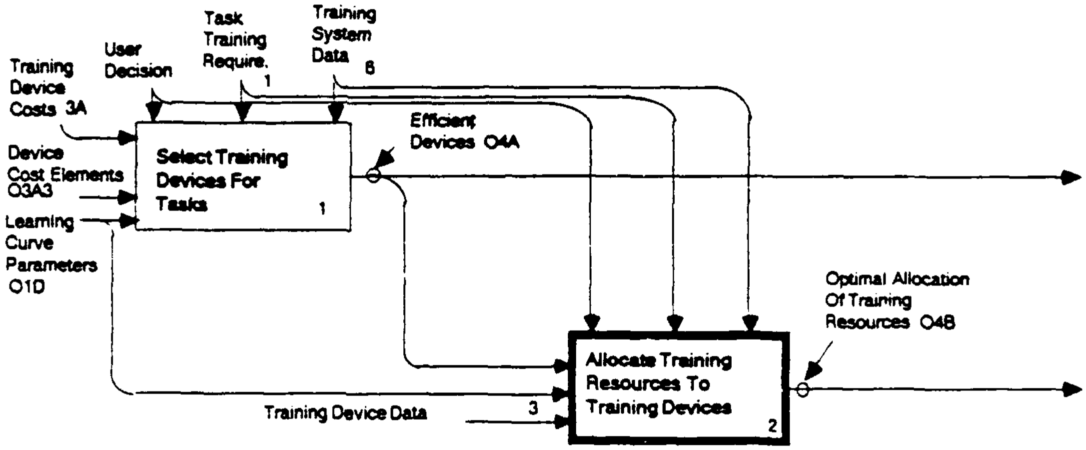
ceiling(argument) = the smallest integer at least as great as the function argument, and where

GRADS = the required annual student throughput (6A1).

<u>OPT/A4133: Determine actual utilization</u>. The annual utilization in hours ($UACT_i$) is calculated by dividing the total number of hours in which the device will be used in a year by the number of simulators, as follows:

$$UACT_i = \frac{DEVT_i * GRADS}{NDEV_i}$$

<u>OPT/A4134: Update model assumptions</u>. In this step, the calculated utilization ($UACT_i$) is compared to the nominal utilizations ($UNOM_i$) assumed in the model. If these utilizations are too different (e.g., 2000 hours versus 3000 hours), the user may decide to revise the assumed simulator utilization. In this case $UNOM_i$ will be assigned the value of $UACT_i$. When the revisions have been made, the analysis will be repeated to obtain a selection based on the revised values.

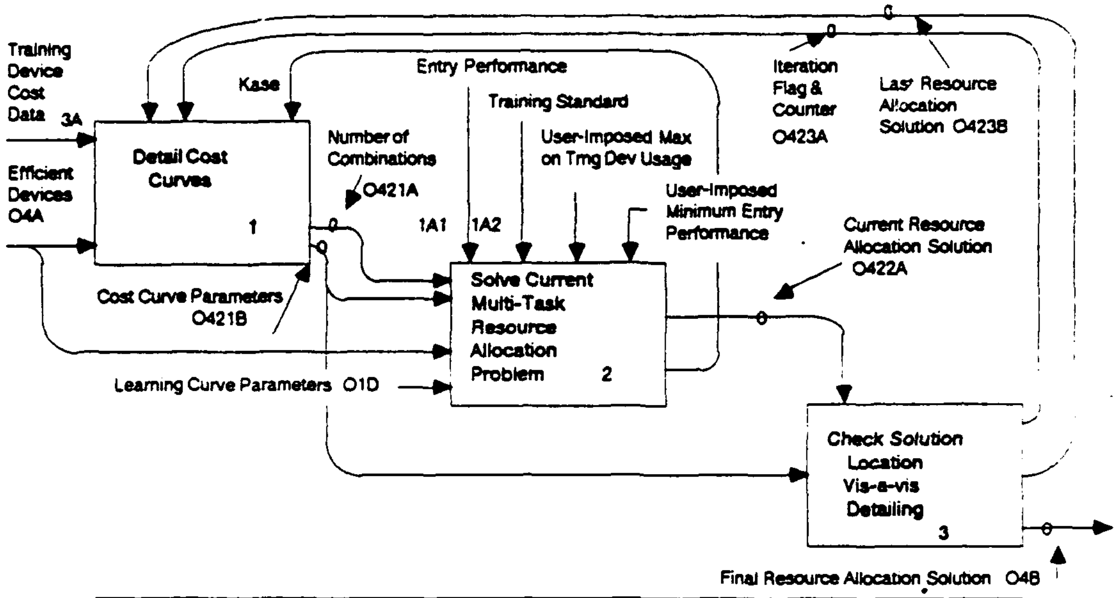


Node: OPT/A4	Title: Assign Training To Devices	Date: 7/12/88	Number: OPT-27

OPT/A42: Allocate Resources to Training Devices

The resource allocation activity determines a refined training program minimizing total system cost to train students to criterion across tasks, subject to constraints on the amount of time a training device may be used or the performance level at which the device may be used. It employs a detailed cost function for each training device.

The algorithm to solve the resource allocation problem is heuristic and iterative. The general idea is to detail the device cost curves only around the solution. The strategy is to begin with relatively undetailed cost curves and generate a first solution using these curves. Then the cost curves are detailed around that solution and the cost minimizing solution is found anew. If the new solution is found to lie within the cost curve domains detailed, the process is terminated, and the solution is deemed optimal. If the current solution does not lie totally within current domains of detailing, the cost curves are redetailed around the current solution, and the cost minimization solution found. If the process is not terminated after a predetermined set of iterations, the process is terminated and the last solution found deemed "optimal", even though it lies outside the domain of detailing for one or more devices. In any



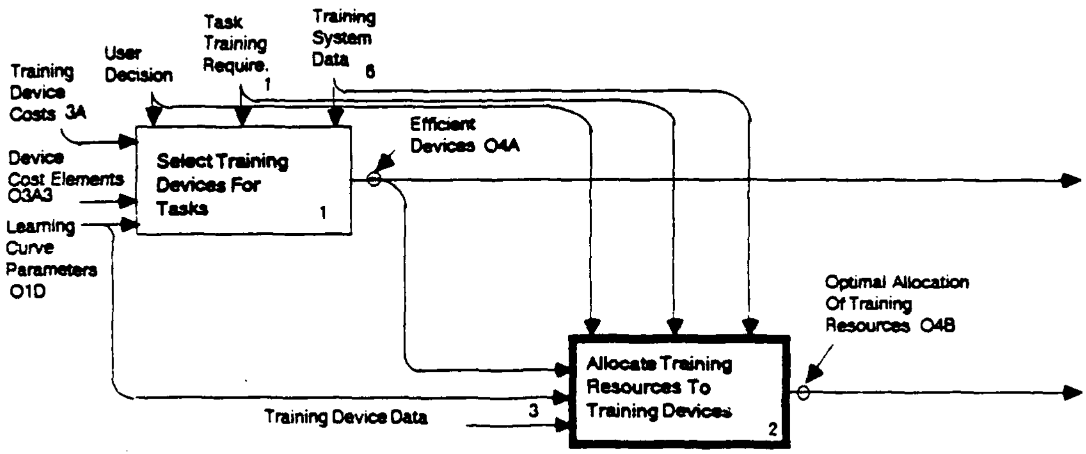
Node: OPT/A42	Title: Allocate Training Resources to Training Devices	Date: 8/1/88	Number: OPT-31

case, the true cost functions are used to determine the per-student cost of training for the final solution.

OPT/A421: Detail Cost Curves. On the first visit to this activity each cost curve is detailed into (at most) a two segment continuous, piecewise linear function. The first segment corresponds to employment of training device assets on hand, and the final segment corresponds to acquisition and employment of new assets. This simple two segment approximation is made to get the iterative solution process going.

On subsequent visits to this activity the cost curves are detailed for several segments on and around the per-student device usage found in the preceding solution. This will generally be a discontinuous piecewise linear curve--a stairstep function with sloping steps.

On each visit to this activity the number of segments detailed must be limited to keep the total number of combinations reasonable, since the optimization problem will be solved for most, if not all, of the cost curve segment combinations. If the number of combinations is too large, then some of the curves are further simplified, at the expense of accuracy. The curves are selected for simplification based on least potential error introduced by the simplification.

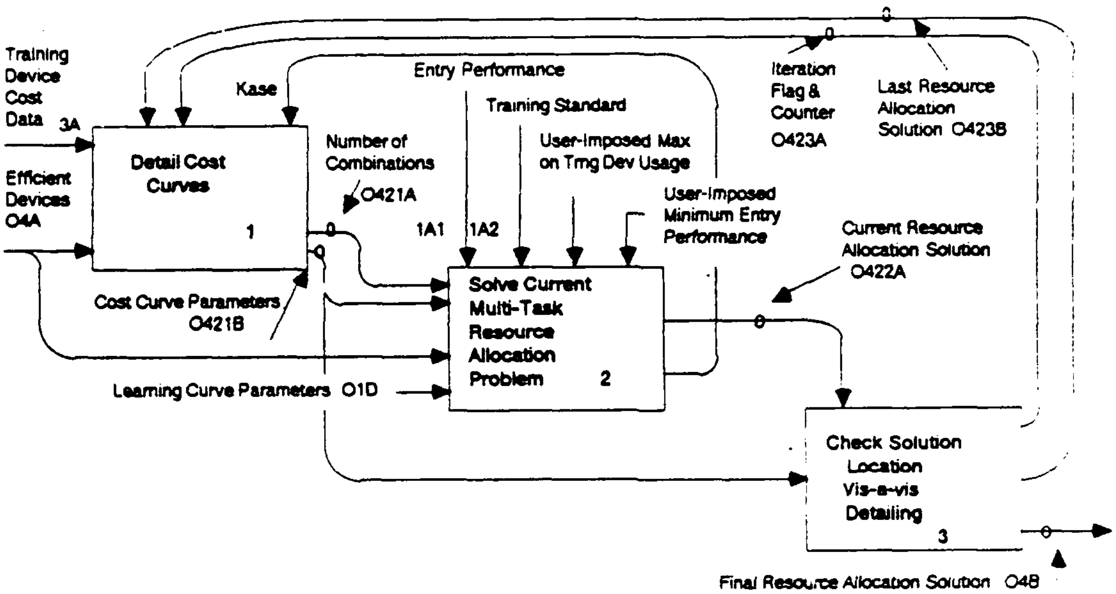


Node: OPT/A4	Title: Assign Training To Devices	Date: 7/12/88	Number: OPT-27

<u>OPT/A422: Solve Current Multi-Task Resource Allocation Problem</u>. On each visit to this activity the constrained multi-task resource allocation problem is solved under the current set of device cost functions detailed in OPT/A421. The solution procedure involves solving the problem for each possible combination of cost curve segments.

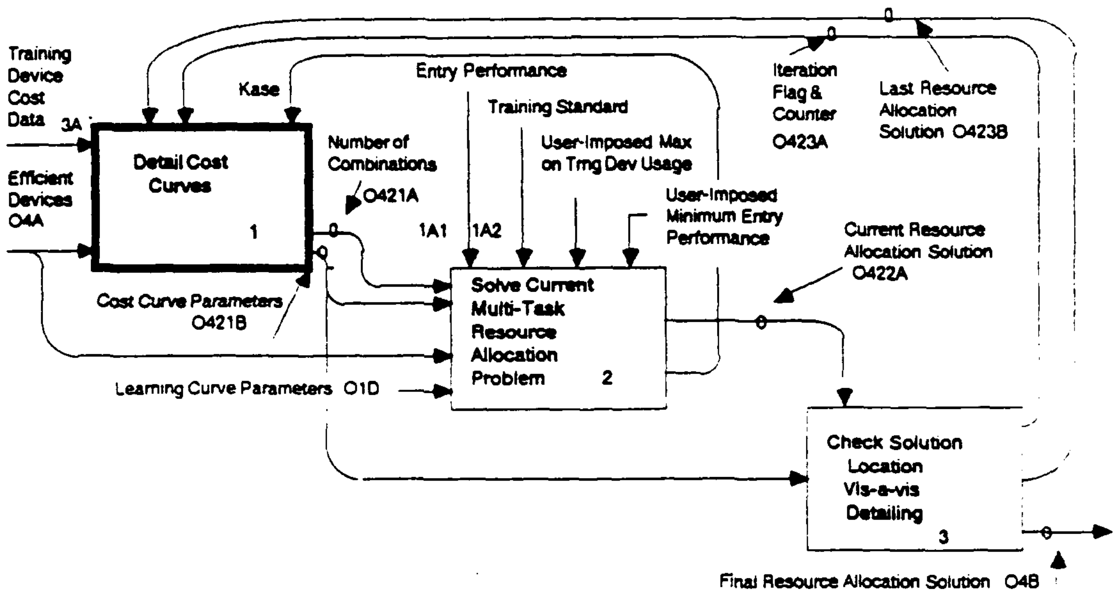
<u>OPT/A423: Check Solution Location Vis-a-Vis Detailing</u>. The solution generated in OPT/A422 will specify a certain total per-student usage for each training device. If these usages fall within the domain of detailing for the respective device cost functions, then the solution has been found with proper consideration of cost curve detailing. If not, then the usage for one or more devices falls outside the domain of detailing, so will not represent desired accuracy. In that case, another iteration through OPT/A421 and OPT/A422 will be conducted, time permitting, in an effort to bring the solution into consonance with the domains of cost curve detailing.

If convergence is not obtained within a mandated number of iterations through OPT/A421 and OPT/A422, then the last solution obtained is used, and the concomitant inaccuracy in cost curve detailing accepted. Even in this case the cost of the final solution is evaluated using the properly detailed device cost



Node:	Title: Allocate Training Resources to	Date:	Number:
OPT/A42	Training Devices	8/1/88	OPT-31

curves. Thus, failing convergence, the solution will still be feasible and accurately costed, but is expected to be somewhat more costly than a truly optimal solution. Experience to date suggests that the optimization error will tend to be small even if convergence is not achieved.



Node:	Title:	Date:	Number:
OPT/A42	Allocate Training Resources to Training Devices	8/1/88	OPT-31

OPT/A421: Detail Cost Curves

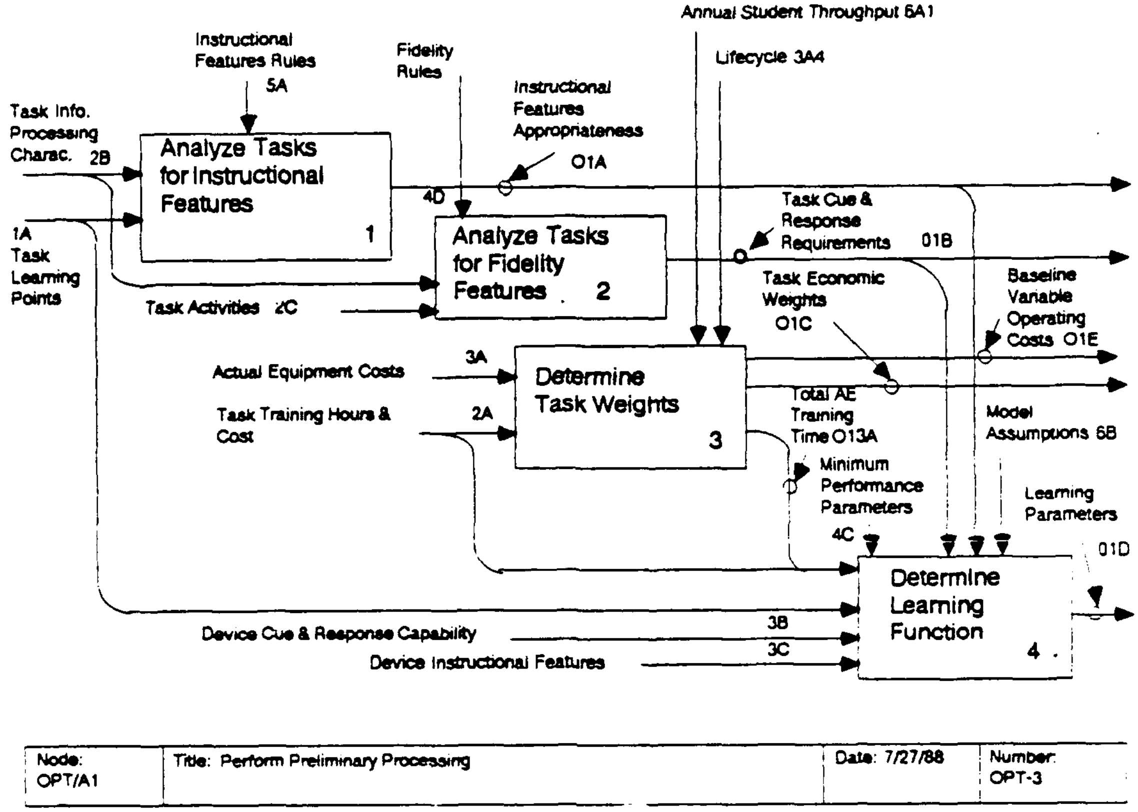
This activity develops a detailed, piecewise linear cost curve for each training device, based on the device cost data (3A). Each curve incorporates the effects of discrete acquisition of devices, but not economies of scale. To facilitate future OSBATS capabilities, provision is also made to incorporate device assets already on hand.

In principle, each cost curve consists of an infinite series of steps, but the optimization algorithm can only handle cost curves with a small number of steps at any one time. So it is necessary to detail the cost curve for only a few steps, and to represent the remainder of the curve by (possibly) one finite linear segment preceding the domain of detailing and one infinite linear segment following it. The cost curve for device i is formally defined by the following parameters:

t = Expected total time that an individual student will use the training device in question during the training program.

$CUMC_i(t)$ = Training device costs attributed to each individual student if each student uses training device for amount of time t.

N_i = Number of segments in the cost curve.



First, a rule base is interrogated to compare task data and rule conditions. The instructional feature rules (5A) associate instructional features with specific task characteristics. An example of such a rule is given below:

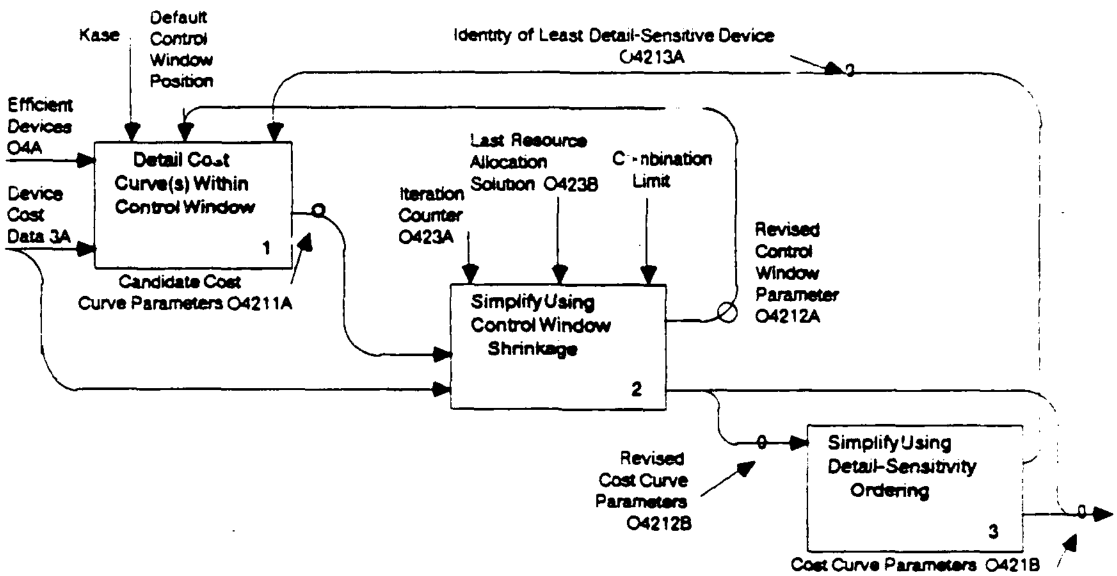
IF: Entry performance (1A1) < 0.4,
and Intrinsic feedback (2B11) is absent,
and The task involves continuous movement (2B1),
or procedures (2B2),
or decision making/rule using (2B4),

THEN: Automated Performance Alerts is indicated for this task.

This activity compares the conditions of the instructional feature rules to the task characteristics (2B) and task learning points (1A), and identifies matches, which it passes on to the next activity.

Next, this activity takes the matches produced in the interrogation, and sets the corresponding cells of IFT_{Tk}. The IFT matrix is defined as follows:

$$IFT_{Tk} = \begin{cases} 1 \text{ if a match was found between feature k and task T} \\ 0 \text{ otherwise.} \end{cases}$$



Node: OPT /A421	Title: Detail Cost Curves	Date: 8/1/88	Number: OPT-32

n = Running index on total number of devices acquired ($n = 0,1,2, \ldots, N-1$)

T_{in} = Starting value for n-th segment of cost curve $CUMC_i(t)$

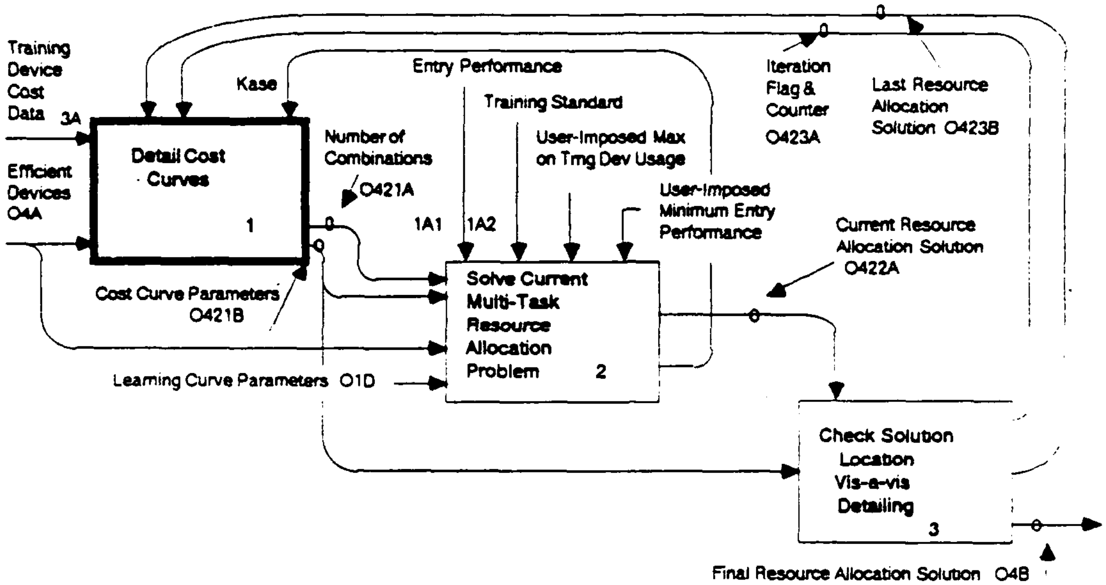
A_{in} = Height of cost curve $CUMC_i(t)$ at beginning of n-th segment.

B_{in} = Slope of n-th segment of cost curve $CUMC_i(t)$.

With two exceptions, each segment is open on the left and closed on the right. The first exception is the first interval, which is a single point $t = 0$. The second exception is the final interval, which is open on the left and extends with constant slope and without limit on the right. The cost curve is defined as follows, as n ranges from 0 to N_i:

$$CUMC_i(t) = \begin{cases} A_{i0} & \text{if } t = T_{i0} \\ A_{in} + B_{in} * (t - T_{in}) & \text{if } T_{in} < t <= T_{in+1} \\ A_{iN_i} + B_{iN_i} * (t - T_{iN_i}) & \text{if } T_{iN_i} < t \end{cases}$$

Discontinuities occur in the cost curve at the points at which the training requires the procurement of an additional

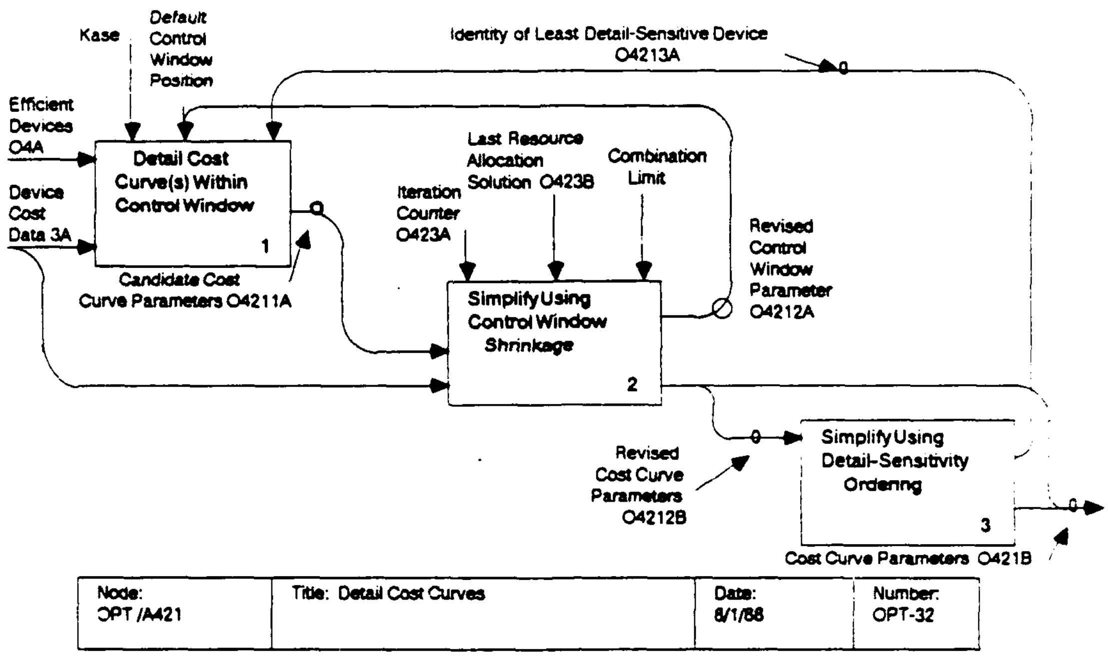


Node: OPT/A42	Title: Allocate Training Resources to Training Devices	Date: 8/1/88	Number: OPT-31

training device. These discontinuities occur at the endpoints (T_{in}) of the cost curve.

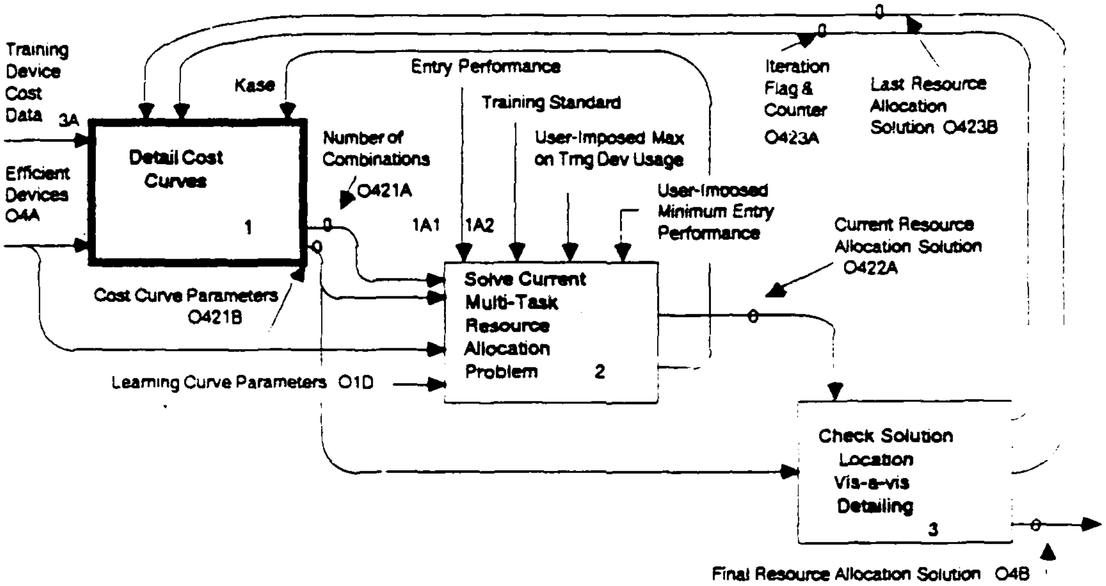
This activity restricts the numbers of segments on the various curves so that the total number of combinations is no more than MAXKB, the maximum number of combinations allowed. This is accomplished by iteratively decreasing the number of segments on the cost curves. The process begins at OPT/A4211 by detailing the cost curves within an initial "window" defined in terms of a domain and range for cost curve detail, and a maximum number of segments allowed. The domain is selected so that the curve will be exact in the region of current interest, i.e., at the last determined value of device usage. If the total number of combinations exceeds MAXKB, an effort is made to reduce the number of segments on the various curves by simply shrinking the "window" within which each curve is detailed (OPT/A4212). If this fails to reduce the total number of combinations sufficiently, then certain curves are simplified to one or two linear segments (OPT/A4213); the curves to be so simplified are chosen to minimize the maximum error potentially introduced by the simplification.

<u>OPT/A4211: Detail Cost Curves, Within Control Window</u>. This activity is controlled by four parameters: KASE, CRANGE, TRANGE, and NRANGE. On the first visit to this activity (KASE = 1) each



cost curve is detailed into (at most) a two segment continuous, piecewise linear function. The first segment corresponds to employment of training device assets on hand, and the final segment corresponds to acquisition and employment of new assets. This simple two segment approximation is made to get the iterative solution process going. On subsequent visits to this activity (KASE = 2) the cost curves are detailed further into segments on and around the per-student device usage found in the preceding solution. The control window is basically defined by CRANGE and TRANGE. CRANGE is the "height" of the window, i.e., the range within which the curve must be detailed; it is expressed in dollars per student. TRANGE is the "width" of the window, i.e., the domain within which the curve must be detailed; it is expressed in hours per student. The parameter NRANGE specifies the number of segments desired, and may be considered another control window parameter.

On the first pass through this activity three important measures of regret are defined for each training device i: $RGRT21_i$ = the increase in possible cost error if a cost curve of a single infinite segment is used instead of a curve with two segments; $RGRT31_i$ = the increase in possible cost error if a cost curve of a single infinite segment is used instead of a curve with three or more segments; and $RGRT32_i$ = the increase in possible cost error if a cost curve of a two segments is used



Node: OPT/A42	Title: Allocate Training Resources to Training Devices	Date: 8/1/88	Number: OPT-31

instead of a curve with three or more segments. These regrets are determined by first calculating ERR1, the maximum error that can occur if the curve is simplified to a single infinite segment, and ERR2, the maximum error that can occur if the curve is simplified to exactly two segments. If the curve has three or more segments, there will be no error for student usage times falling in the middle segments. If there are no assets on hand, then ERR1 equals the step height,

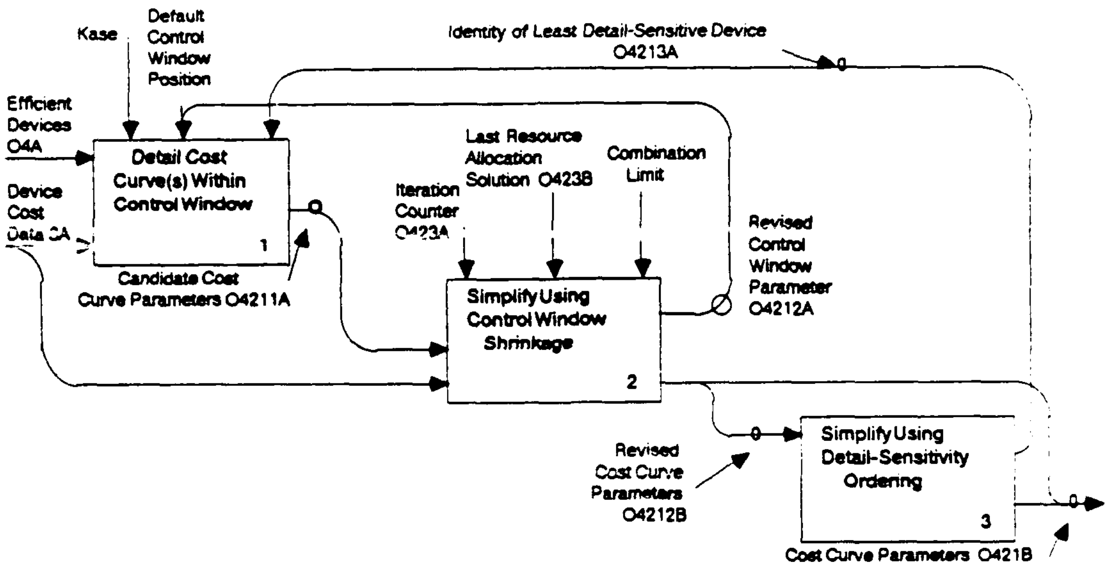
$$STEPHT_i = (INV_i/LC_i + FOC_i)/GRADS \ .$$

Since curves of 2 segments offer no advantage over curves of 1 segment when there are no assets on hand, ERR2 is set to infinity. If there are assets on hand, then ERR1 is computed based on a tentative two-segment detailing:

$$ERR1 = \text{maximum } (STEPHT_i - (B_{i2}-B_{i1})*T_{i2},\ (B_{i2}-B_{i1})*T_{i2}).$$

In the case of assets on hand, $ERR_2 = STEPHT_i$.

Once the error terms determined, then the (potential) regret in going from a curve with 2 segments to a curve with 1 segment is $RGRT21_i = ERR1 - ERR2$. The regret in going from a curve with 3 or more segments to one with 2 segments is $RGRT32_i = ERR2$ since there is no error if curves of 3 or more segments are used and

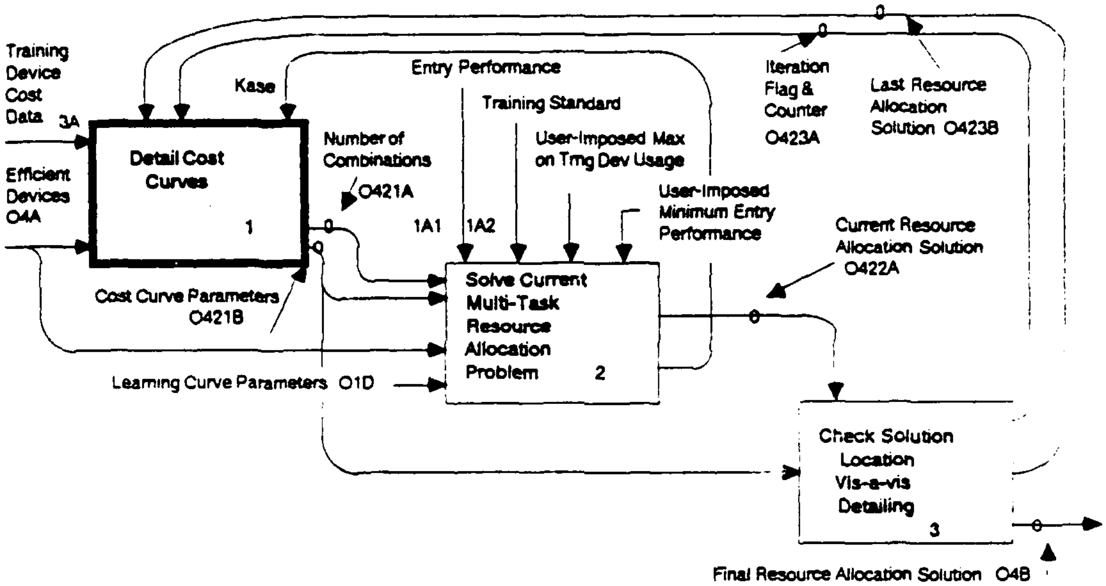


Node: OPT /A421	Title: Detail Cost Curves	Date: 8/1/88	Number: OPT-32

student usage falls in the middle segment(s), a condition that will be forced by the optimization algorithm. Similarly, $RGRT31_i$ = ERR1.

OPT/A4212: Simplify Using Control Window Shrinkage. The number of combinations of device cost curve segments is determined as $NCOMBO = NTAB_1 * NTAB_2 * \ldots * NTAB_{ND}$ where $NTAB_i$ is the number of segments in the cost curve for device i, and where ND is the number of training devices under consideration. If NCOMBO exceeds MAXKB, then the size of the control window is reduced by 25% and OPT/A4212 is revisited to detail the cost curves again within the smaller window. More precisely, the following substitutions are made: CRANGE = 0.75 * CRANGE, TRANGE = 0.75 * TRANGE, and NRANGE = 0.75 * NRANGE (integer truncation). If NRANGE is reduced to zero, then this activity is terminated and control is transferred to OPT/A4213 to continue the simplification process.

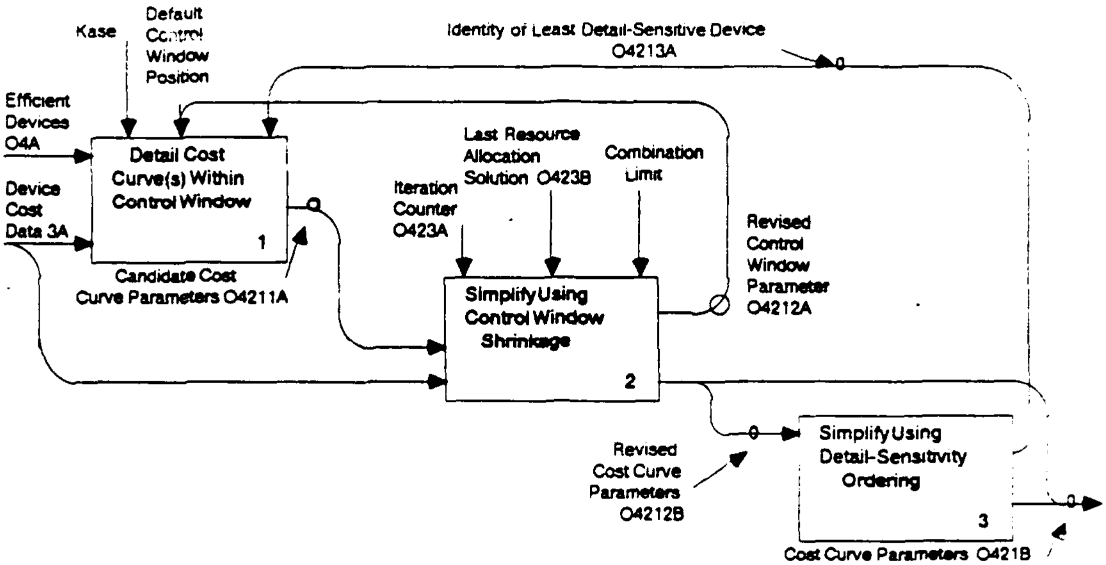
OPT/A4213: Simplify Using Detail-Sensitivity Ordering. This activity is invoked if OPT/A4212 fails to reduce the number of combinations of device cost curve segments to an acceptable level. It is then necessary to simplify one or more of the device cost curves to only one or two segments. Such simplification will generally cause a loss of accuracy in the cost computation. In order to minimize the impact of such



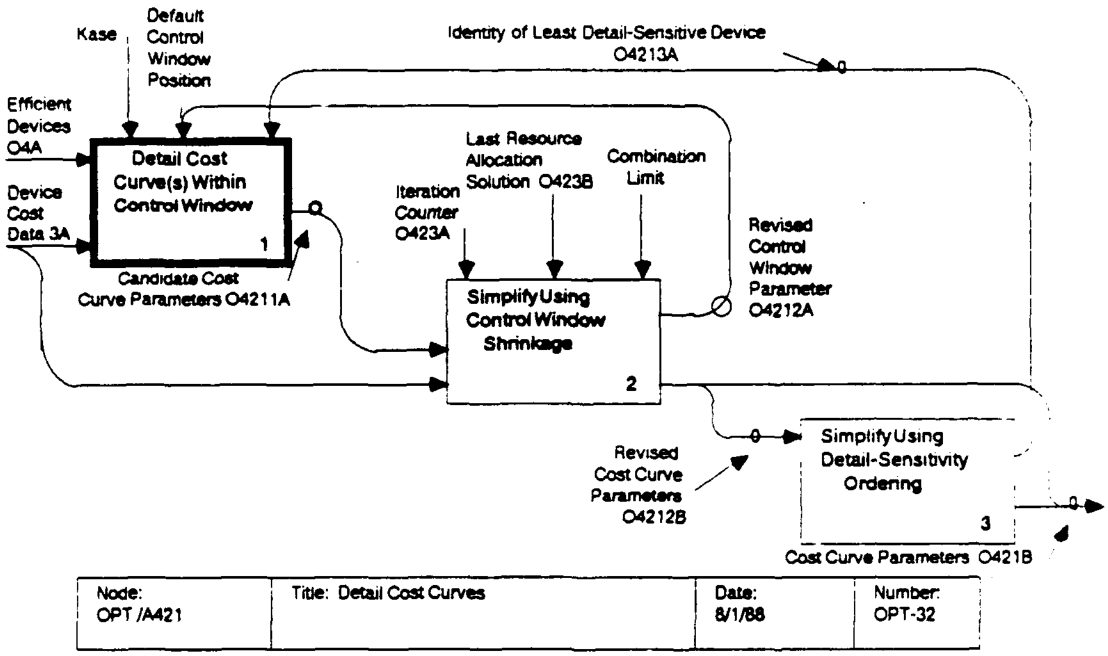
Node:	Title:	Date:	Number:
OPT/A42	Allocate Training Resources to Training Devices	8/1/88	OPT-31

simplification, the curves to be simplified are chosen, in turn, to introduce minimum regret, based on the measures $RGRT21_i$, $RGRT31_i$, and $RGRT32_i$ computed in OPT/A4211.

Once the curve to be simplified next is identified, together with the desired number of segments, the simplification is accomplished by repeating (if necessary) the KASE = 1 computations of OPT/A4211. This yields a one or two segment curve. If a two segment curve is constructed, but a one segment curve is desired, the curve is simplified to one segment by substituting B_{i2} for B_{i1}.



Node: OPT /A421	Title: Detail Cost Curves	Date: 8/1/88	Number: OPT-32

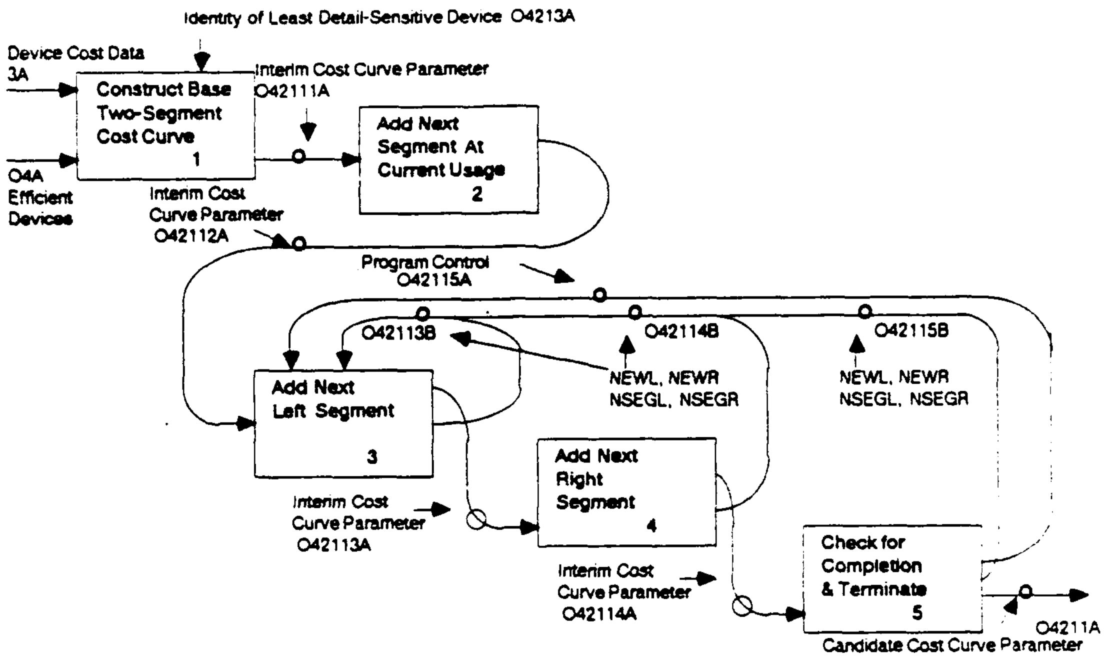


OPT/A4211: Detail Cost Curves Within Control Window

This activity begins by constructing a one- or two-segment curve accounting for assets on hand, if any. Then, if further detailing is desired (KASE=2), a segment is added on top of the current per-student usage time. Additional segments are added to the left and right of this segment, until the desired number of segments (NRANGE) is achieved or the detailing exceeds one of the control window dimensions (TRANGE, CRANGE). The control window parameters are treated by this activity as guidelines more than strict upper bounds. They are not used to impose a Procrustean simplification to the cost curve.

The k-th cost curve segment for device i is defined by a starting time T_{ik}, a height A_{ik}, and a slope B_{ik}. There is no need to define the end of the segment since it simply occurs when the next segment begins.

OPT/A42111: Construct Base Two-Segment Cost Curve. This activity begins by constructing the first segment, anchored at the origin, so that $T_{i1} = 0$ and $A_{i1} = 0$. If there are assets on hand, then the slope of the first segment is $B_{i1} = FOC_i/UMAX_i + VHRC_i$, which assumes a pro rata per-student share of fixed operating costs, together with variable operating costs, but excludes investment costs, since the assets are already owned.



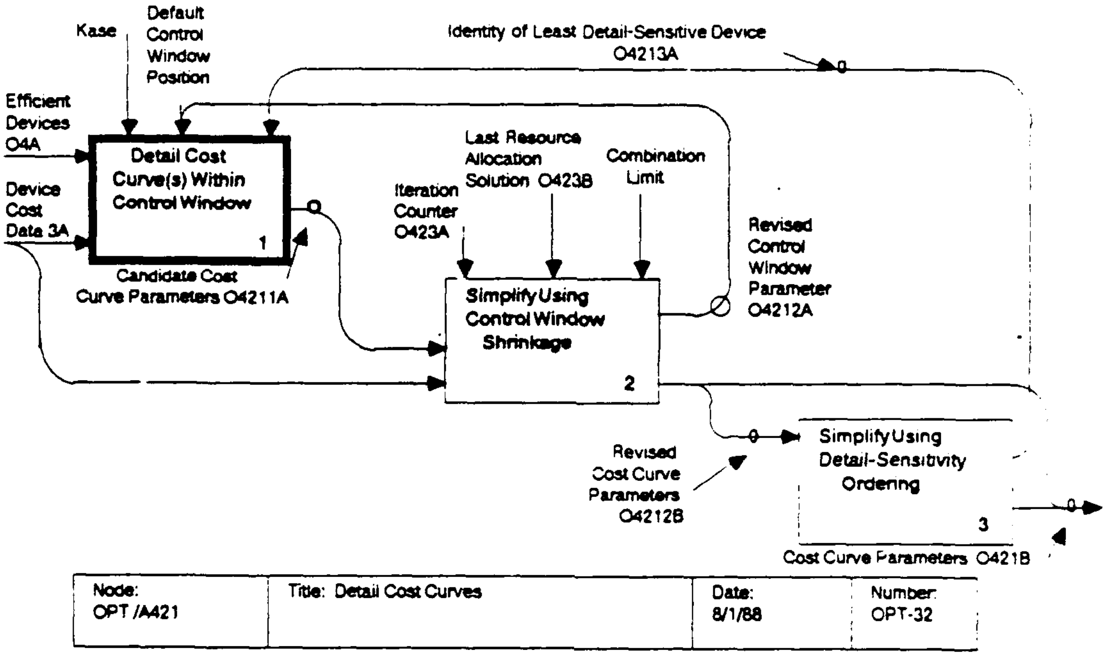
Node: OPT/A4211	Title: Detail Cost Curves Within Control Window	Date: 8/1/88	Number: OPT-33

If there are no assets on hand, then this first summary segment includes investment costs and is calculated as

B_{i1} = $(INV_iC/LC_i + FOC_i)/UMAX_i + VHRC_i$.

A second segment is constructed only if there are assets on hand. In that case, the beginning of the second segment is found as T_{i2} = $ASSETS_i*UMAX_i/GRADS$, i.e., the utilization at which assets on hand are fully used. The beginning of the second segment is set to $A_{i2} = A_{i1} + B_{i1}*(T_2 - T_1)$, i.e., the height of the first segment at its end. The slope of the second segment is then set to $B_{i2} = (INV_i/LC_i + FOC_i)/UMAX_i + VHRC_i$.

<u>OPT/A42112: Add Next Segment At Current Usage</u>. If additional detailing is desired (KASE=2), then that segment for the ideal curve that is located atop the current device usage $TTOT_i$ is next spliced into the base curve. To do this it is necessary to determine the number of <u>new</u> devices associated with $TTOT_i$: NEW = $TTOT_i*GRADS/UMAX_i - ASSETS_i$, rounded up to the nearest integer. If NEW = 0, then no new devices need be acquired to satisfy $TTOT_i$, so the base curve is adequate as it stands and no detailing is required. However, the domain and range of detailing are marked by the endpoints of the first segment, (TLEFT,CLEFT) = (T_{i1}, A_{i1}), and (TRIGHT,CRIGHT) = $(T_{i2}, A_{i1}+B_{i1}*(UMAX_i/GRADS)$. Also, two parameters, NSEGL and



NSEGR, marking the leftmost and rightmost deliberately detailed segment, are initialized equal to 1. Two other parameters, NEWL and NEWR, indicating the number of new devices associated with these two segments, are initialized equal to 0.

If NEW > 0 then a segment, tentatively numbered N_i+1, will be added. (N_i is either 1 or 2, depending on whether $ASSETS_i = 0$ or $ASSETS_i > 0$, respectively.) This is done by redefining $N_i = N_i+1$ and setting

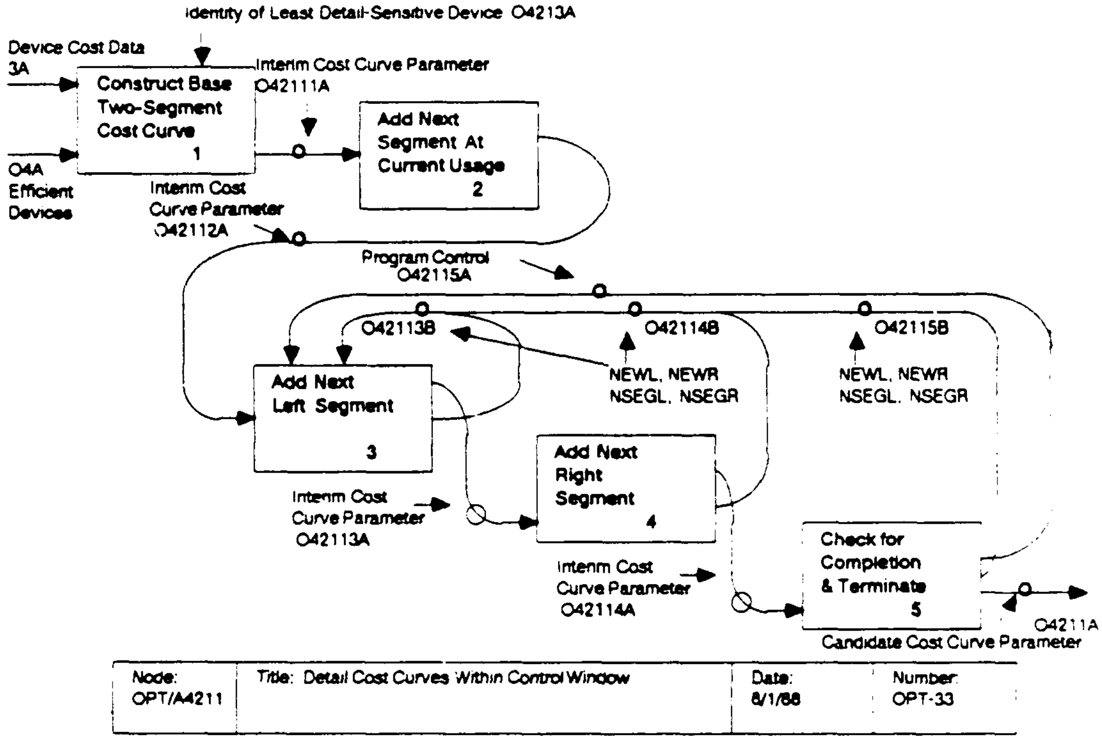
$$T_{iN_i} = (ASSETS_i + NEW - 1)*UMAX_i/GRADS$$

$$A_{iN_i} = A_{iN_i-1} + B_{iN_i-1} * (T_{iN_i} - T_{iN_i-1}) + (INV_i/LC_i + FOC_i)/GRADS$$

$$B_{iN_i} = VHRC_i \; .$$

The domain and range of detailing is determined as the segment endpoints: $(TLEFT, CLEFT) = (T_{iN_i}, A_{iN_i})$ and

$(TRIGHT, CRIGHT) = (TLEFT+UMAX_i/GRADS, A_{iN_i} + B_{iN_i} * UMAX_i/GRADS)$.



If NEW > 1, then the segment N_i-1 that was previously the last segment needs to be retained. But if NEW = 1 and $ASSETS_i > 0$, this segment is dispensable, since the new segment N_i begins before or at the same point as that segment; in that case segment N_i-1 is replaced in toto by segment N_i, and N_i is redefined as N_i-1. However, if NEW = 1 and $ASSETS_i = 0$, the segment N_i-1 (which is anchored at the origin) is retained so that the cost curve will give a zero cost in response to a zero per-student time, rather than the value $STEPHT_i > 0$.

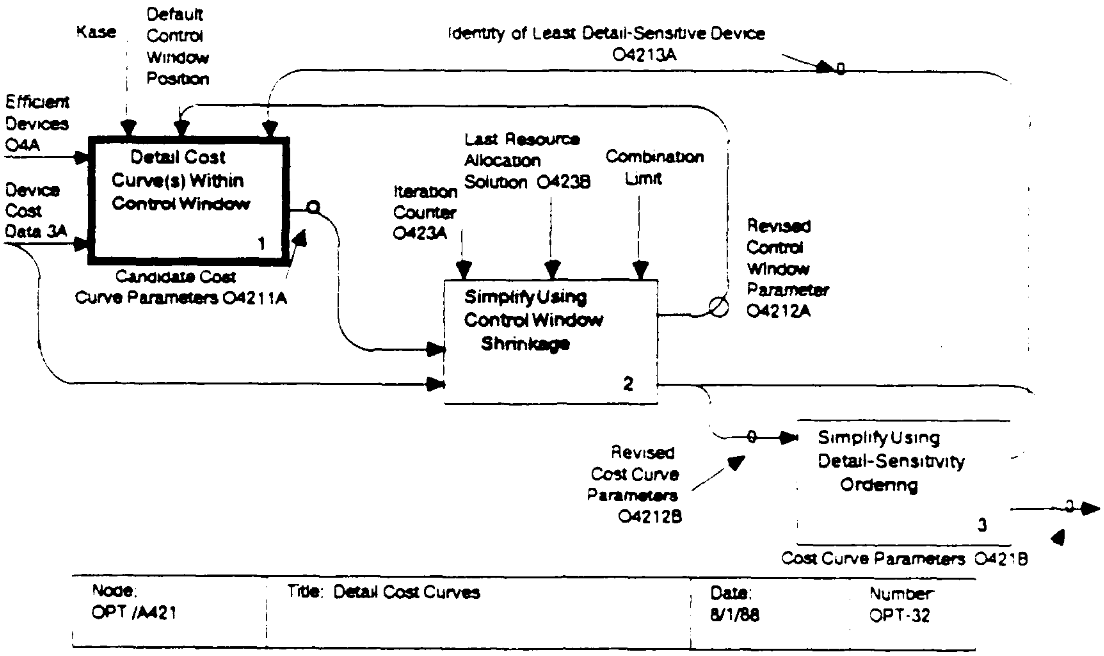
Upon completion of the detailing of the segment N_i at the current usage, NSEGL and NSEGR are initialized equal to N_i, and NEWL and NEWR are initialized equal to NEW. Then a new final infinite segment is added by setting

$$N_i = N_i + 1$$

$$T_{iN_i} = TRIGHT$$

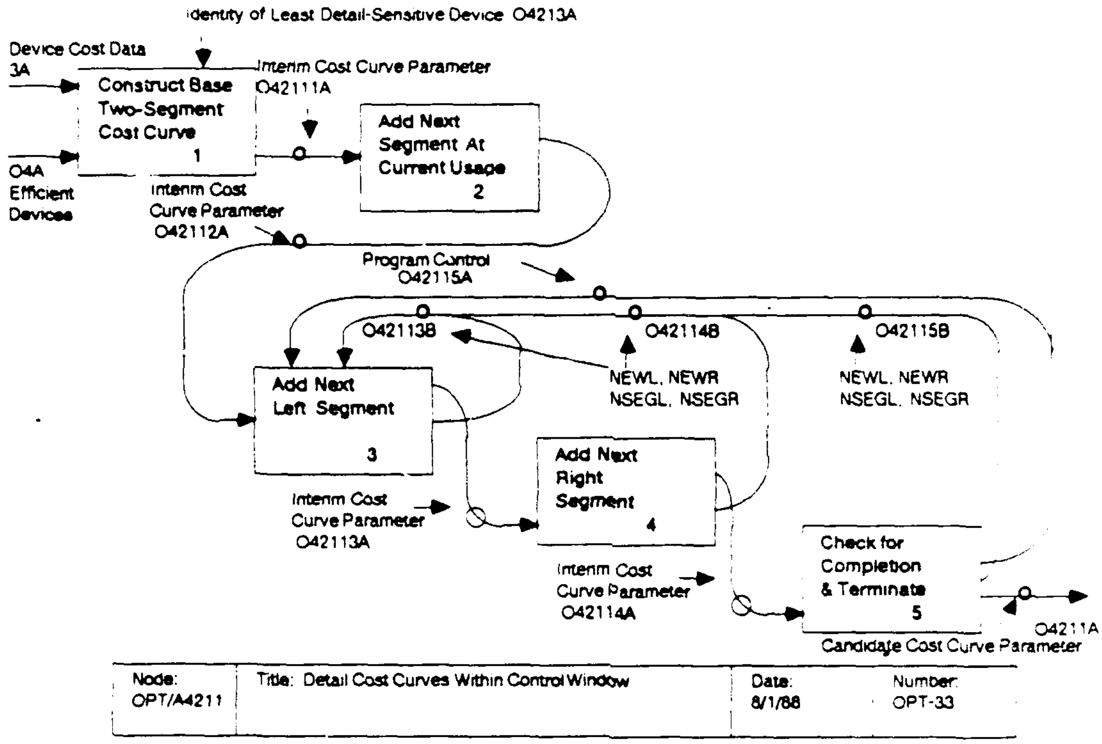
$$A_{iN_i} = CRIGHT$$

$$B_{iN_i} = (INV_i/LC_i + FOC_i)/UMAX_i + VHRC_i \ .$$



<u>OPT/A42113: Add New Left Segment</u>. At this point a check is made to see if further detailing is required, based on the number of segments, the time domain covered, or the cost range covered. In particular, if NSEGL > 1 and NSEGR - NSEGL + 3 < NRANGE, or if NSEGL = 1 and NSEGL - NSEGR + 2 < NRANGE, then the number of detailed segments is deemed deficient. If TRIGHT - TLEFT < TRANGE, then the time domain detailed is deemed deficient. If CRIGHT - CLEFT < CRANGE, then the cost range detailed is deemed deficient. If these tests indicate that further detailing is required, a new segment is detailed to the left of the current detailed domain. However, if detailing a new right segment would cause $TTOT_i$ to fall more near to the center of [TLEFT, TRIGHT], then this step is skipped at this time.

This detailing is begun by setting NEW = NEWL-1; NEW will be the number of new devices corresponding to the segment being detailed. If NEW = 0, then no more detailing is possible to the left, so nothing is done to the curve. However, the control parameters corresponding to the leftmost detailed segment are updated, provided it is not the first segment of the curve. In particular, If NSEGL > 1 then NSEGL is set equal to NSEGL - 1, TLEFT is set equal to T_{iNSEGL} (for the redefined NSEGL), and CLEFT is set equal to A_{iNSEGL}.



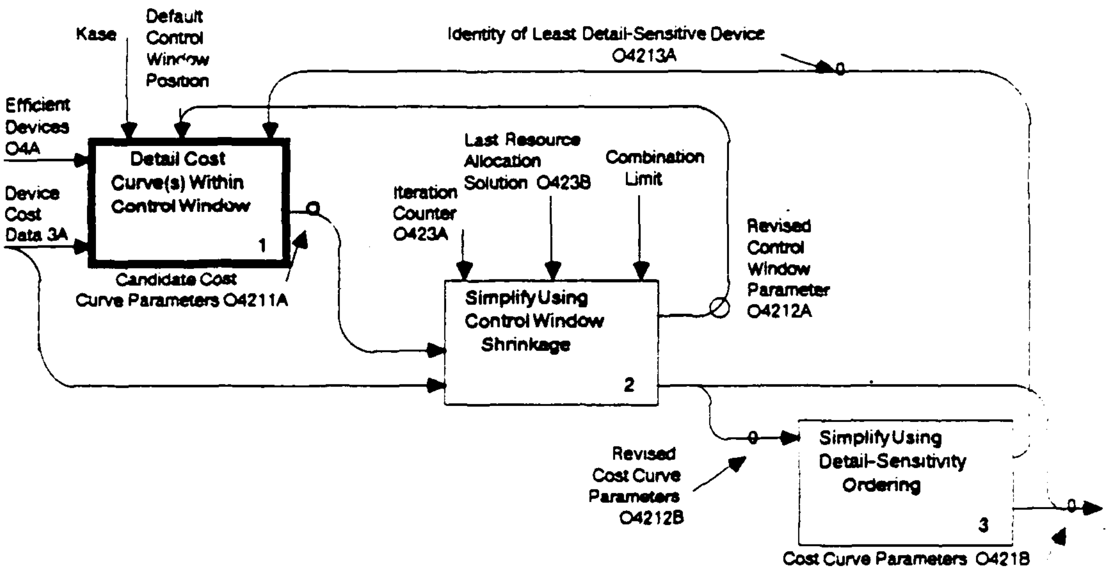
If NEW > 0, then there is room for another segment corresponding to new device acquisitions on the left. The parameter NEWL is set equal to NEW. A new segment will be added at the old left segment index, NSEGL, unless NEW = 1 and NSEGL = 3, in which case the old segment number 2 at the beginning of the post-fixed-assets portion of the base curve is replaced. In that special case of NEW = 1 and NSEGL = 3, the left segment index is shifted to NSEGL = NSEGL - 1 to force the old segment to be replaced by the operations described in the next paragraph.

If NEW > 0 and either NEW is unequal to 1 or NSEGL is unequal to 3, then the segments to the right of NSEGL are renumbered by, essentially, adding 1 to the corresponding indices. Also N_i is set to $N_i + 1$ and NSEGR is set to NSEGR + 1. The segment NSEGL is then redefined by

$$T_{iNSEGL} = (ASSETS_i + NEWL - 1) * UMAX_i/GRADS)$$

$$A_{iNSEGL} = A_{i(NSEGL-1)} + B_{i(NSEGL-1)} * (T_{iNSEGL} - T_{i(NSEGL-1)}) + (INV_i/LC_i + FOC_i)/GRADS \quad .$$

$$B_{iNSEGL} = VHRC_i \quad .$$



Node: OPT /A421	Title: Detail Cost Curves	Date: 8/1/88	Number: OPT-32

TLEFT and CLEFT are set to T_{iNSEGL} and A_{iNSEGL}, respectively.

<u>OPT/A42114: Add New Right Segment</u>. At this point a check is made to see if further detailing is required, using the same logic as at the beginning of OPT/A42113. The actual segment addition is more straightforward on the right, since there are none of the complications associated with the origin or transition from assets on hand to new acquisitions. The actual addition of the segment is accomplished by setting NEW = NEWR+1, NEWR = NEW, NSEGR = NSEGR+1,

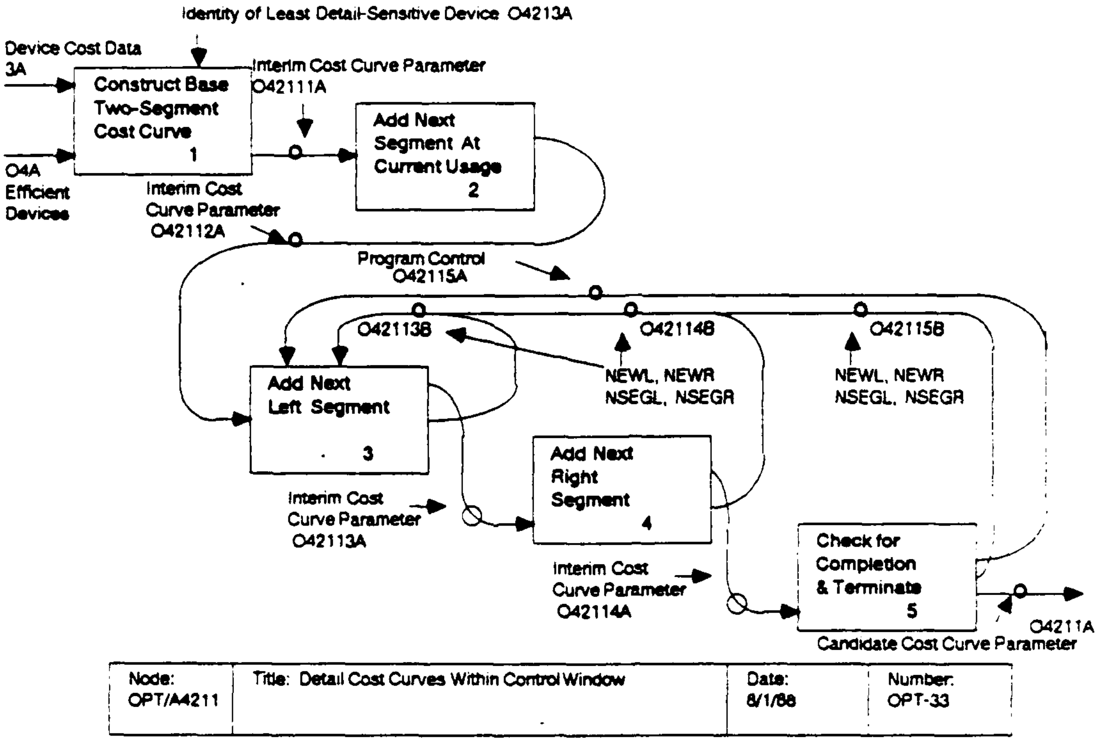
$$T_{iNSEGR} = (ASSETS_i + NEWR - 1) * UMAX_i/GRADS)$$

$$A_{iNSEGR} = A_{i(NSEGR-1)} + B_{i(NSEGR-1)} * (T_{iNSEGR} - T_{i(NSEGR-1)}) + (INV_i/LC_i + FOC_i)/GRADS \quad .$$

$$B_{iNSEGR} = VHRC_i \quad .$$

$$TRIGHT = T_{iNSEGR} + UMAX_i/GRADS$$

$$CRIGHT = A_{iNSEGR} + B_{iNSEGR} * UMAX_i/GRADS \quad .$$



Then a new final segment is added by setting

$$N_i = N_i + 1$$

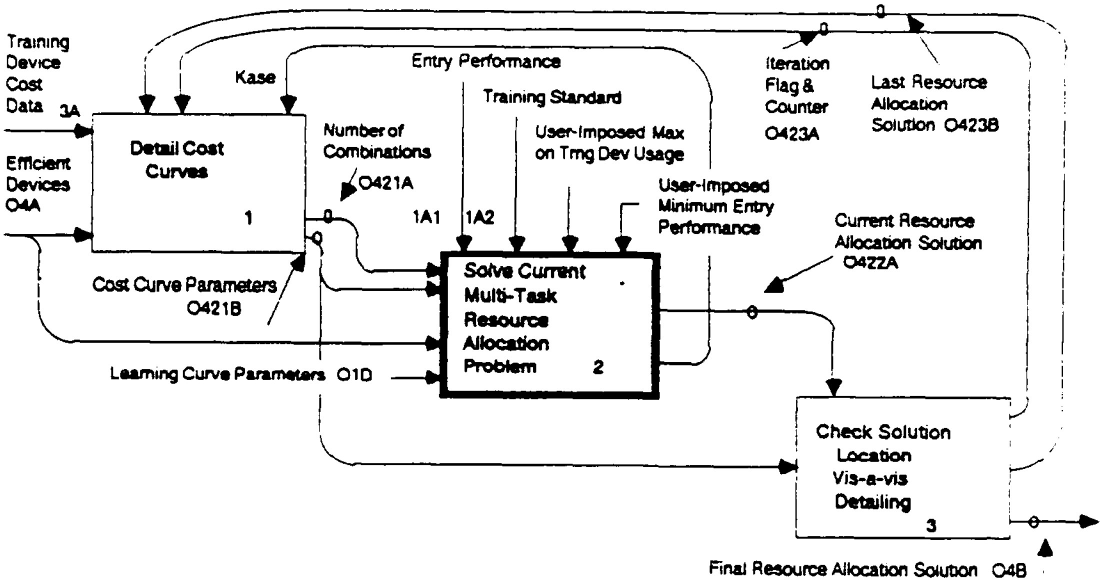
$$T_{iN_i} = \text{TRIGHT}$$

$$A_{iN_i} = \text{CRIGHT}$$

$$B_{iN_i} = (INV_i/LC_i + FOC_i)/UMAX_i + VHRC_i \quad .$$

<u>OPT/A42115: Check for Completion and Terminate</u>. At this point another check is made to see if further detailing is required, using the same logic as at the beginning of OPT/A42113. If further detailing is required, control is transferred to OPT/A42113 to begin another cycle of left-right segment detailing.

If sufficient detailing is accomplished, a check is made to see if there are segments lying to the left of the detailed domain, i.e., if NSEGL > 1. If so, those segments are removed in order to reduce the number of segments in the curve.

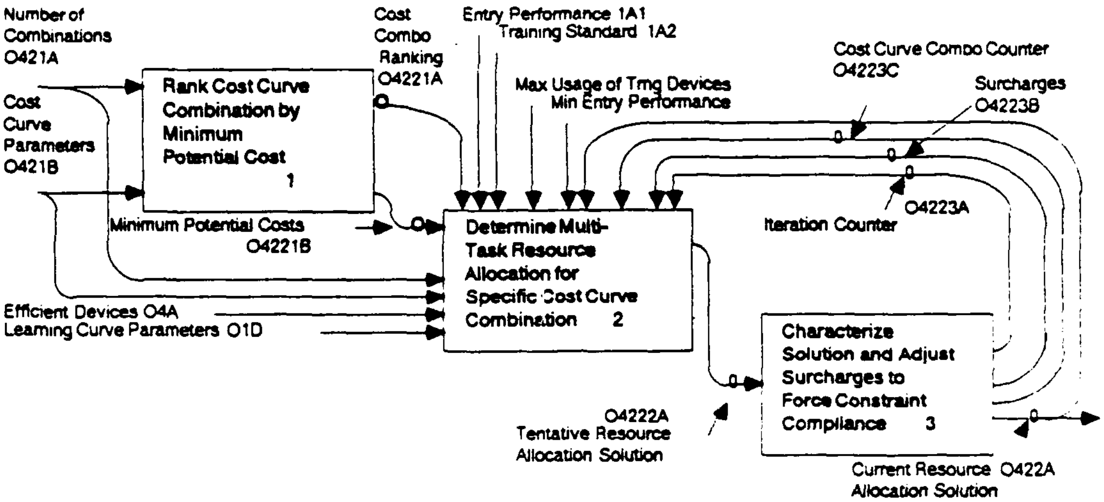


Node: OPT/A42	Title: Allocate Training Resources to Training Devices	Date: 8/1/88	Number: OPT-31

OPT/A422: Solve Multi-Task Resource Allocation Problem.

The minimum cost resource allocation problem is solved in this activity, given the currently detailed cost curves. The discontinuous cost curves obviate an elegant, one-shot solution procedure. Instead, the minimum cost resource allocation problem is solved, in principle, for each possible combination of cost-curve segments. The solution generated for a given combination is then evaluated using the true cost curves. That solution yielding the least total cost is deemed the optimal solution for the current set of cost curves.

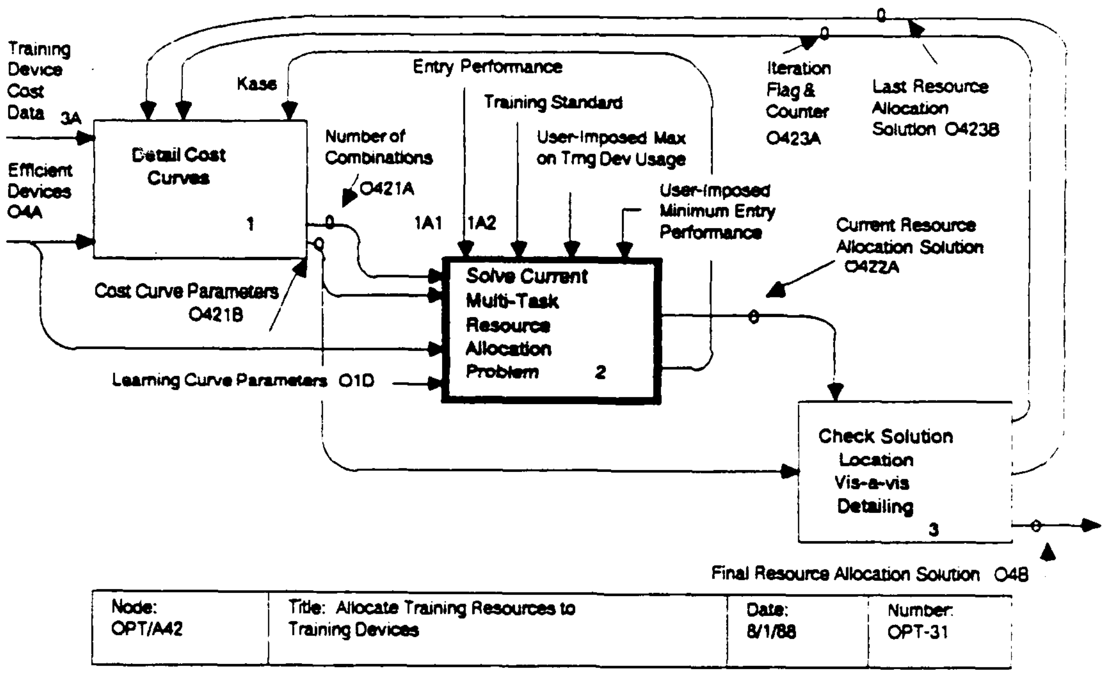
Although the minimum cost resource allocation problem is, in principle, solved for each combination of cost curve segments, it is sometimes possible to determine that a given combination will not possibly yield a superior solution. This is accomplished by examining the least potential cost associated with the given cost curve combination. If that cost exceeds the minimum cost solution found up to that point, then the associated cost curve combination is ruled out of consideration. Further, cost curve combinations are examined in order of increasing minimum potential costs. If any combination is deemed excessively costly compared to the incumbent solution, then all subsequent cost curve combinations are then known to be excessively costly, too, so processing for the current cost curve detailing is terminated.



Node:	Title:	Date:	Number:
OPT/A422	Solve Multi-Task Resource Allocation Problem	8/1/88	OPT-34

(This is an application of the branch-and-bound technique of operations research.) The required ranking of cost curves by minimum potential cost is accomplished at OPT/A4221.

For a given combination of cost curve segments, the multiple task constrained resource allocation problem is solved by iterating between nodes OPT/A4222 and OPT/A4223. A tentative solution is generated at OPT/A4222, assuming no constraints on device usage. The mandated device usage is then examined at OPT/A4223, in light of actual device usage constraints. If device usage constraints are violated, then artificial surcharges are applied to the slopes of the current cost curve segments, making the overused devices appear more expensive, and the problem is solved again at OPT/A4222. After each iteration it is determined whether the solution respects prescribed usage constraints and the surcharges satisfy associated optimality conditions. If so, the solution is deemed viable and optimal for the current cost curve segments. Otherwise, iteration continues. If no satisfactory solution is found after a prescribed set of iterations, or if a suspicious "flip-flopping" behavior is observed in the surcharges, then the particular cost curve segment combination is deemed infeasible and is abandoned. (This use of surcharges to force a solution within prescribed constraints is an application of the sequential unconstrained minimization technique (SUMT) of operations research).

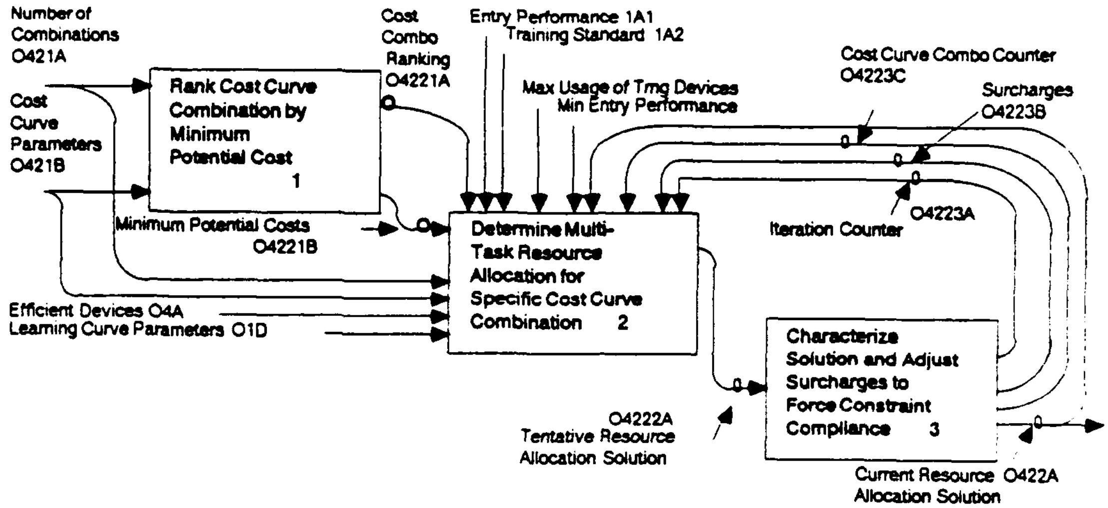


<u>OPT/A4221: Rank Cost Curve Combinations by Minimum Potential Costs</u>. The lower bound CLB_{KOMBO} on possible per-student cost is found for each combination, indexed KOMBO, of cost curve segments by summing the minimum cost for the active segment for each device cost curve:

$$CLB_{KOMBO} = \sum_i A_{ik_i}$$

where k_i is the cost curve segment being used for device i under combination KOMBO. A ranking array KBRANK(KB) is then determined by sorting the combinations by increasing cost. Then when KB=1, KBRANK(KB) is the number of the combination having the minimum value of least potential cost, based simply on the summed lower edges. Similarly, KBRANK(NKOMBO) would be the number of the combination having the greatest value of least potential cost. It is important to understand that "least potential cost" refers simply to the least cost that could be associated with the given segments, and that there is yet no implication on the feasibility of training effectively at the associated device usages.

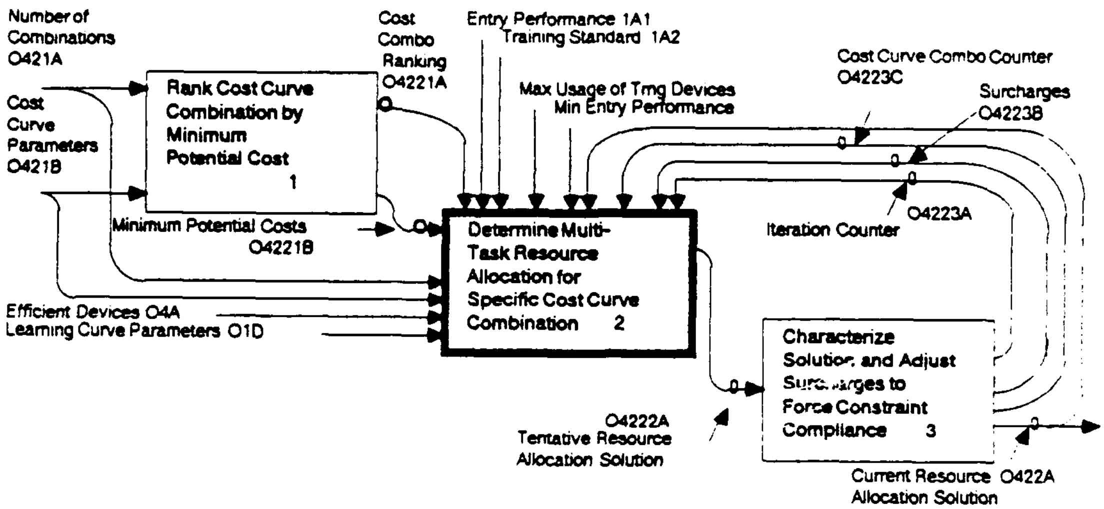
<u>OPT/A4222: Determine Multi-Task Resource Allocation for Specific Cost Curve Combination</u>. This activity involves first setting up a set of single task problems, using current



Node: OPT/A422	Title: Solve Multi-Task Resource Allocation Problem	Date: 8/1/88	Number: OPT-34

performance and cost data, user-imposed constraints, and cost curve segment surcharges (OPT/A42221), and then solving these single task problems, in sequence (OPT/A42222). For each device i the total allocation $TTOT_i$ is found as the sum of the allocations $TRNT_{iT}$ over the tasks, assuming each single-task problem is soluble.

<u>OPT/A4223: Characterize Solution and Adjust Surcharges to Force Constraint Compliance</u>. At this activity the complete multiple task solution is checked for constraint violations and surcharges are adjusted correspondingly (OPT/A42231). Then a check is made to see if the solution and surcharges satisfy conditions allowing declaration of a solution for the multiple task resource allocation problem, or whether further iteration or abandonment are in order (OPT/A42232).



Node: OPT/A422	Title: Solve Multi-Task Resource Allocation Problem	Date: 8/1/88	Number: OPT-34

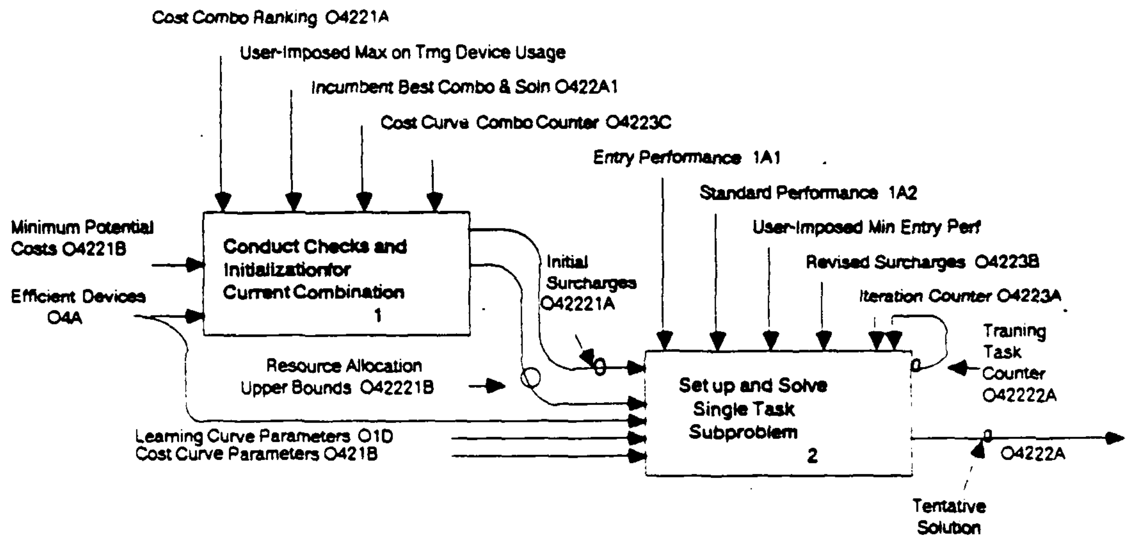
<u>OPT/A4222: Determine Multi-Task Resource Allocation for Specific Cost Curve Combination</u>.

<u>OPT/A42221: Conduct Checks and Initializations for Current Combination</u>. The minimum potential cost, CLB_{KOMBO}, associated with combination KOMBO = KBRANK(KB) is compared to CBEST, the per-student cost of training under the incumbent best solution (CBEST is initialized as infinity). If CLB_{KOMBO} exceeds CBEST, then KOMBO can not possibly yield a better solution, nor can any of the following combinations, associated with KB+1, KB+2, etc.; so in that case OPT/A422 processing is terminated.

Otherwise, combination KOMBO is processed. For each training device, i, the maximum allowed per-student device usage is set equal to the minimum of the user-imposed usage, $USMAX_i$, and the usage corresponding to the end of the assumed cost curve segment, LEV_i, for the device. I.e.,

$$TAVAIL_i = \text{Minimum}\ (USMAX_i,\ T_{i(LEV_i+1)})\ ,$$

where the end of the LEV_i segment is determined by the beginning of the LEV_i+1 segment. This constraint is applied as an upper bound to device i allocations for each task T, also: UB_{iT} =



Node: OPT/A4222	Title: Determine Multi-Task Resource Allocation for Specific Problem	Date: 8/1/88	Number: OPT-35

$TAVAIL_i$. (This constraint will not prevent overallocation to device i, since allocations for several tasks may add to a violation, but it prevents overallocation for a single task.)

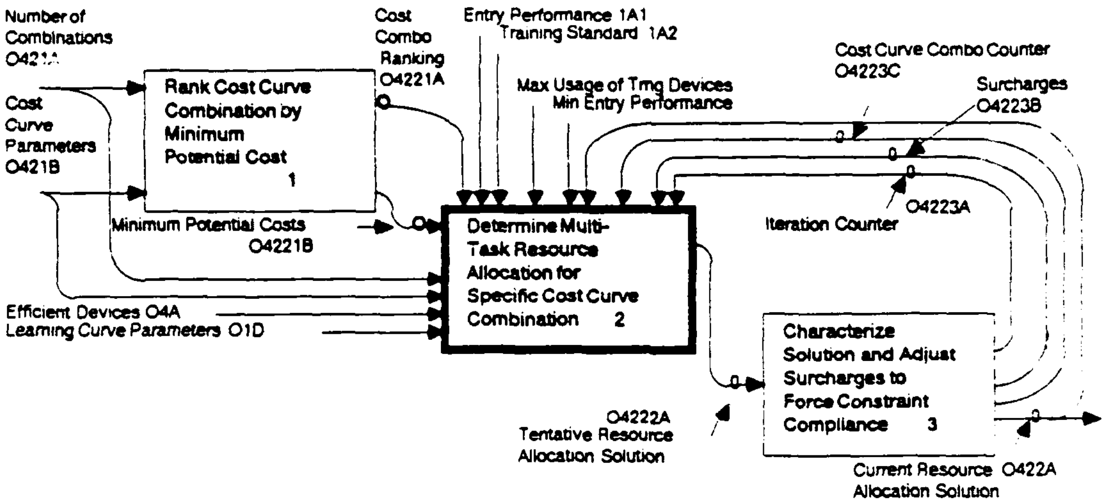
Also in this activity the cost surcharge $SURCHG_i$ for each device i is initialized at 1.

<u>OPT/A42222: Set Up and Solve Single-Task Subproblems</u>. On each visit to this activity an effective price for each device i is set by applying the current surcharge $SURCHG_i$:

$$B_i = SURCHG_i * B_{iLEV_i} \quad .$$

On the first visit to this activity each surcharge has the initialized value $SURCHG_i = 1$, corresponding to no penalty for usage. On subsequent visits to this activity, as iteration is accomplished to force overall device usage constraint accommodation, the adjusted surcharges are used, as set in OPT/A42231.

At this point a simple single-task constrained resource allocation problem is defined for each task:



Node: OPT/A422	Title: Solve Multi-Task Resource Allocation Problem	Date: 8/1/88	Number: OPT-34

Find a sequence of devices, $IDV_1, \ldots, IDV_k, \ldots, IDV_{NSTAGE_T}$ and associated per-student training allocations $TRNT_{IDV_1,T}, \ldots,$ $TRNT_{IDV_k,T}, \ldots, TRNT_{IDV_{NSTAGE_T},T}$ that will minimize

$$\text{Total Cost} = \sum_i B_i * TRNT_{iT}$$

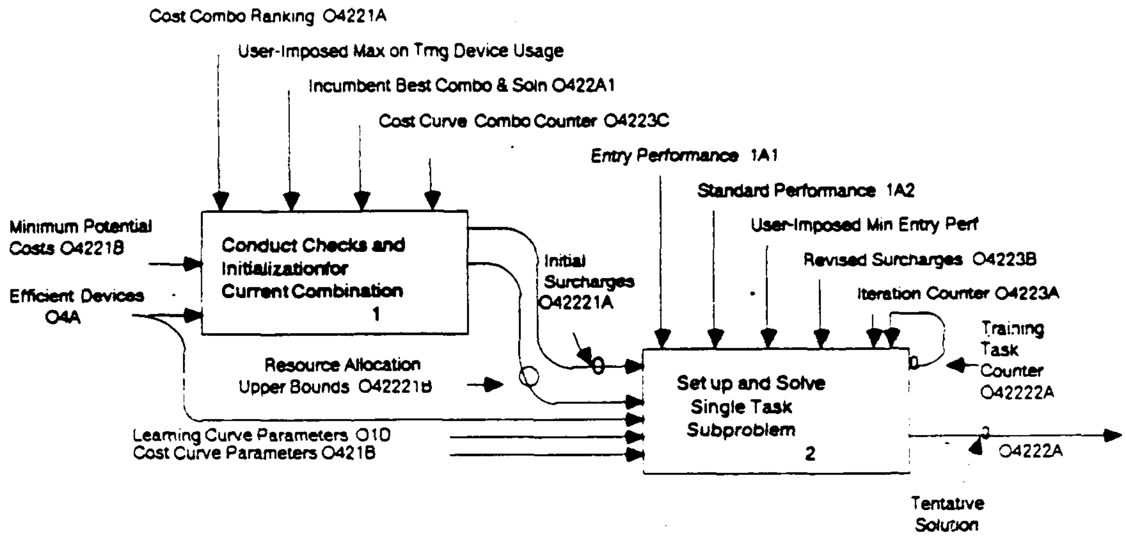
subject to

$$\sum_i TRNT_{iT} \leqq UB_{iT}$$

$$\begin{array}{l}\text{Student Performance}\\ \text{on task T prior to}\\ \text{training on device } i\end{array} \geqq EPMIN_{iT}$$

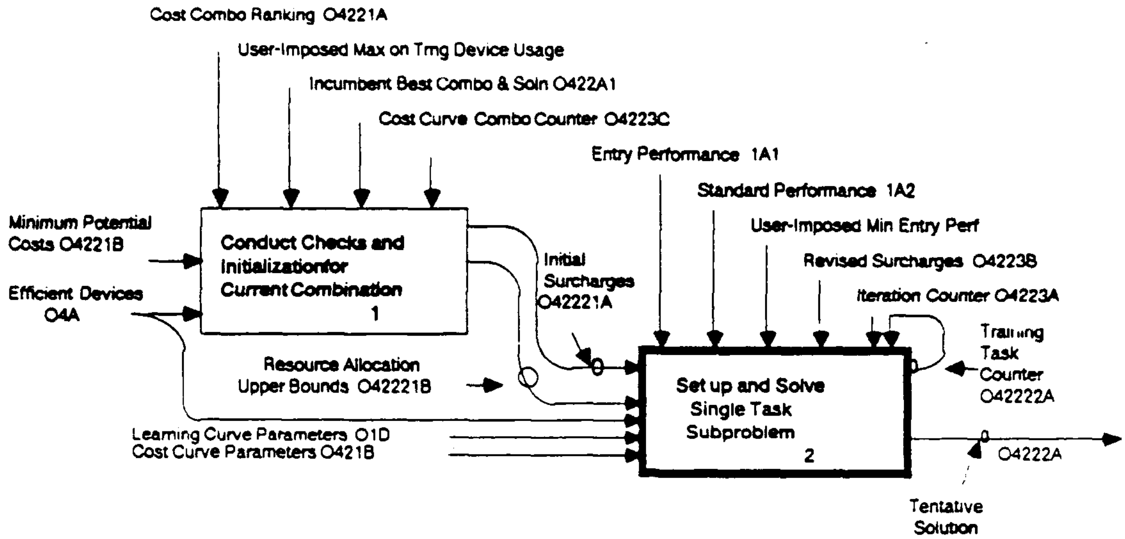
where student entry performance to the task is $PENTRY_T$ and where student performance at the end of training on each device is determined by the learning curve for that device on that task.

The actual single-task optimization algorithm is based on mathematical analysis to determine optimal switching between devices in an unconstrained environment, and iterative techniques



Node: OPT/A4222	Title: Determine Multi-Task Resource Allocation for Specific Problem	Date: 8/1/88	Number: OPT-35

to lock in allocations and proceed to a simplified problem in order to handle the allocation and performance constraints. The algorithm is detailed in OPT/A422221 - OPT/A422225.



Node: OPT/A4222	Title: Determine Multi-Task Resource Allocation for Specific Problem	Date: 8/1/88	Number: OPT-35

OPT/A42222: Set Up and Solve Single-Task Subproblems

In order to simplify this discussion it is convenient to recapitulate the precise problem definition, but suppressing any reference to the task T, which is an implicit index on all variables, as indicated in the OPT/A4222 discussion of this activity. Let

i, j = indexes referring to the training devices

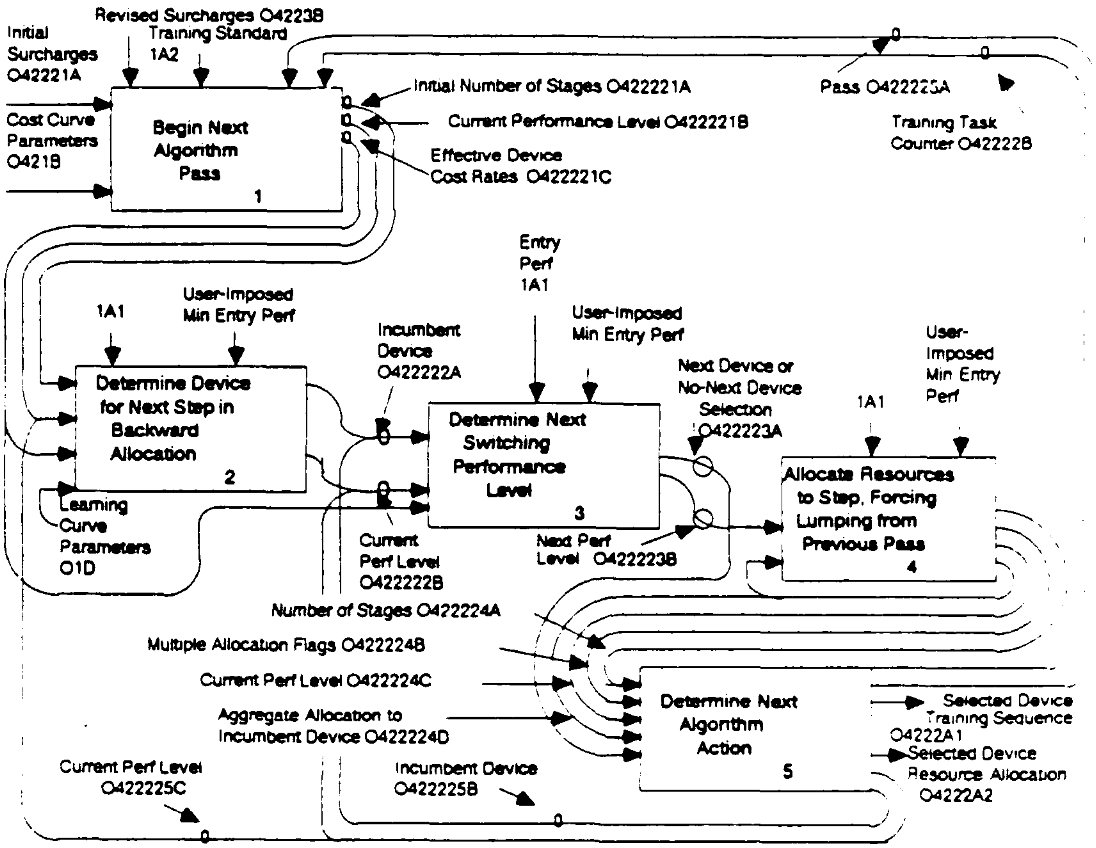
$TRNT_i$ = allocation of per-student time to device i

B_i = Attributed per-student variable cost rate for training on device

p_{j_k} = performance at completion of k-th stage training, on device j_k

UB_j = user imposed upper bound on per-student training time for device j.

$PERF_i(TRNT_i, P_{now})$ = performance achieved on training device i after training an amount of time $TRNT_i$, given an entry level of performance P_{now}.



Node:	Title:	Date:	Number
OPT/A42222	Set up and Solve Single Task Subproblem	8/1/88	OPT-36

The function $PERF_i(t,P_{now})$ is evaluated using the associated inverse function $PINV_i(p)$ as follows:

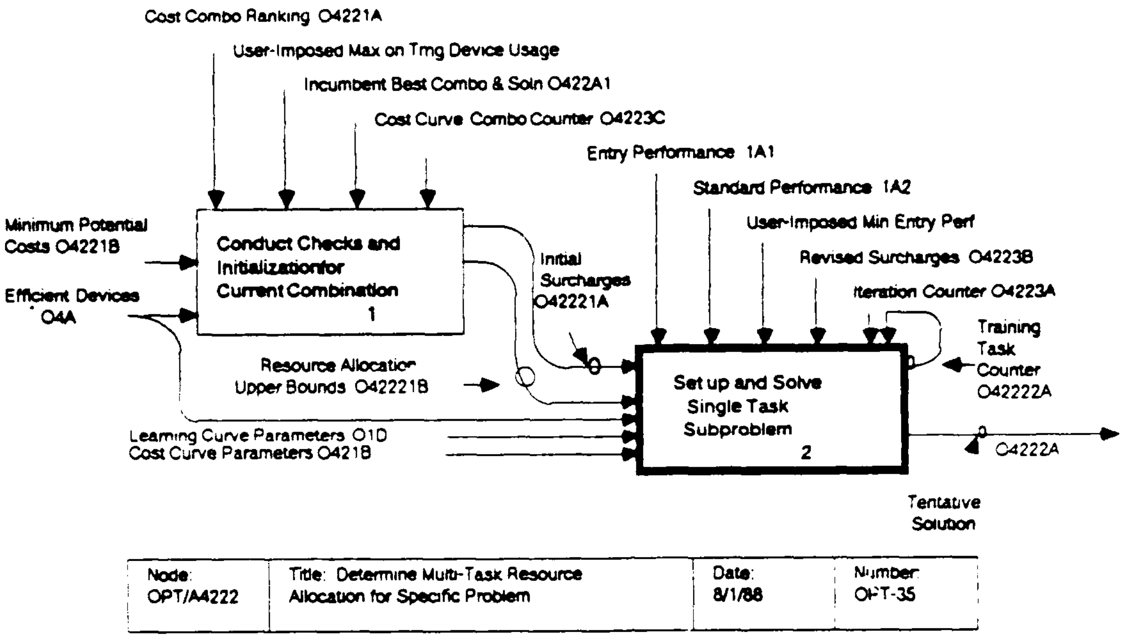
$$PINV_i(p) = \frac{(1 - p/ASM_i)^{1/R} - 1}{TM_i * TSF} .$$

The first step in evaluating $PERF_i(t,P_{now})$ is then to determine the "head start" time parameter associated with the entry performance P_{now}: $HS = PINV_i(PENTRY)$. Then $PERF_i(t,P_{now})$ is calculated as

$$PERF_i(t, P_{now}) = ASM_i \left[1 - \left[1 + TM_i * TSF * (t + HS)\right]^{-R}\right] .$$

With these definitions in hand, the single-task optimization problem becomes:

Determine a sequence $\{IDV_k\}_{k=1,NSTAGE}$ of training devices and associated per-student training allocations $\{TRNT_{IDV_k}\}$ to



$$\text{Minimize} \sum_{k=1,NSTAGE} B_{IDV_k} * TRNT_{IDV_k}$$

subject to allocation constraints

$$TRNT_{IDV_k} \leq UB_{IDV_k}$$

and subject to performance constraints and relations

$$PERF_{IDV_{NSTAGE}}(TRNT_{IDV_{NSTAGE}}, p_{IDV_{NSTAGE-1}}) = P_{STD}$$

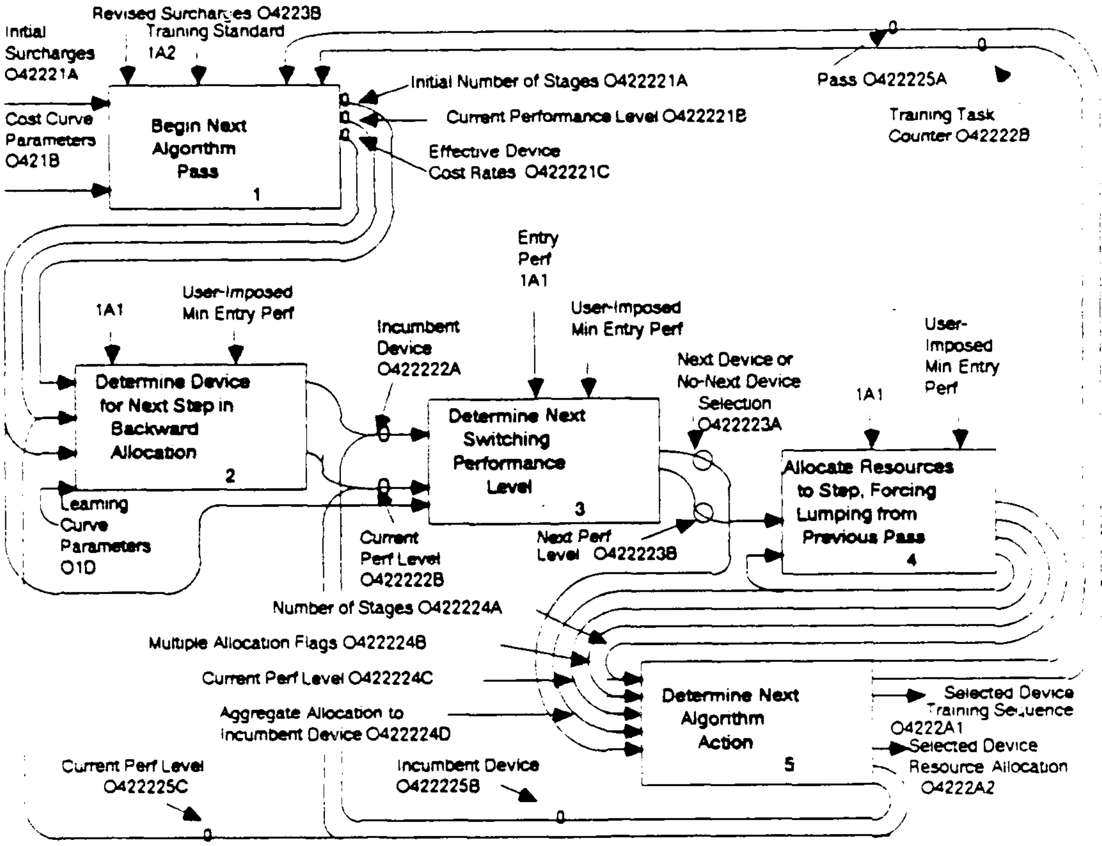
where $p_{j_{N-1}}$ and the other device j_k completion performance levels p_{j_k} are defined recursively by

$$p_{j_k} = PERF_{j_k}(TRNT_{j_k}, p_{j_{k-1}})$$

and where the entry performance to device j_1 is set at the student entry performance level:

$$p_{j_0} = P_{ENTRY}$$

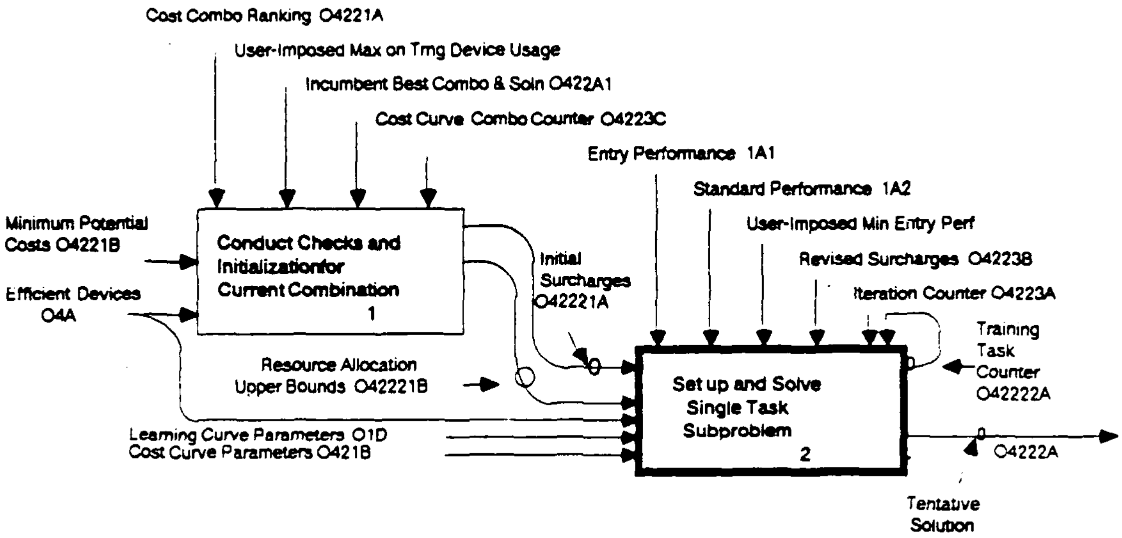
The algorithm may be considered a superset of the training device selection procedures defined in OPT/A41, and shares many



Node: OPT/A42222	Title: Set up and Solve Single Task Subproblem	Date: 8/1/88	Number: OPT-38

of the underlying equations. In fact, an identical solution will be obtained for an unconstrained problem. However, the present algorithm involves more complex heuristic procedures to handle constraints. It contrasts with OPT/A41 by allocating "from the top down", i.e., by first determining that training device to be used on the final stage of student training, then determining that device to be used on the penultimate stage, and so on. (This "backwards" allocation was found preferable for handling performance-and resource-constraints.) More than one pass may be required in order to avoid using the same device at two distinct stages of training, an eventuality that will not be allowed. A tentative selection of a device at more than one point may occur if it must be used in an inefficient manner, in terms of learning per dollar, because a more efficient device is proscribed by constraints. In such a case, the training resources assigned at different stages to the same device on one pass are "lumped" together on the next pass in a forced allocation at that point where the device in question is first selected for employment.

OPT/A422221: Begin Next Algorithm Pass. Processing begins by initializing all allocations to zero, and setting the current performance level to the standard, P_{now} = STD, in preparation for the "backwards" allocation procedure. The pass counter, PASS, is incremented, and the index of the incumbent device, INCMB, is set to a value flagging it as unassigned.

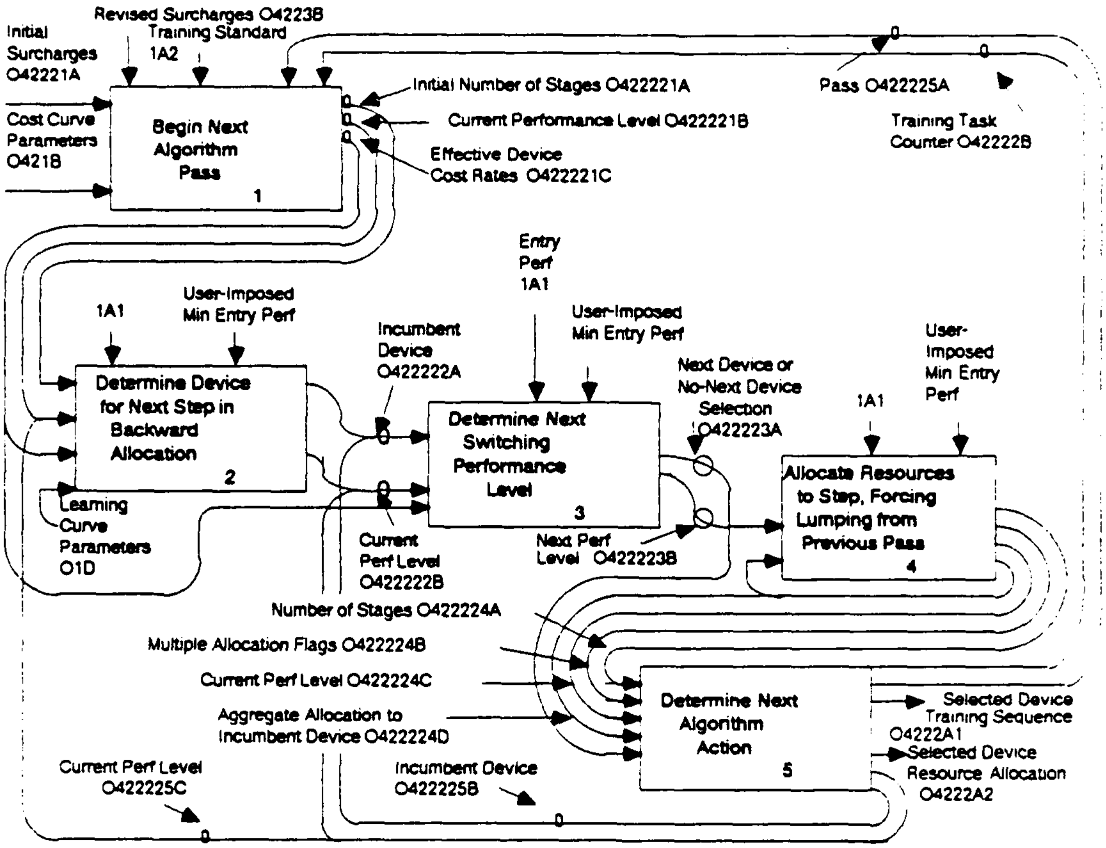


Node: OPT/A4222	Title: Determine Multi-Task Resource Allocation for Specific Problem	Date: 8/1/88	Number: OPT-35

<u>OPT/A422222: Determine Device for Next Step in Backward Allocation</u>. The device to be used for training up to the current performance level, P_{now}, is selected as that qualified device providing the greatest rate of increase in performance with respect to per-student dollar spent at that performance level. For device i this rate is given by

$$PRATE_i = \left[\frac{ASM_i * r * TM_i * TSF}{B_i}\right] \left[1 - \frac{P_{now}}{ASM_i}\right]^{\frac{r+1}{r}} .$$

where ASM_i, r, TM_i, TSF are the asymptote, shape parameter, time multiplier, and time scaling factor for the learning curve for device i (for task T), and where B_i is the effective per-student price for usage of device i. A device is deemed qualified for consideration if it is not the incumbent device, if it can actually be used to train to level P_{now} (i.e., $ASM_i > P_{now}$), if there are appreciable training resources left (i.e., $TNRT_i > UB_i$), and if the current performance level exceeds the minimum allowable entry performance level for the task in question (i.e., $P_{now} > EPMIN_i$). If no qualified devices are available, the problem is deemed insolvable, and OPT/A42222 is abandoned, with a



Node: OPT/A42222	Title: Set up and Solve Single Task Subproblem	Date: 8/1/88	Number: OPT-36

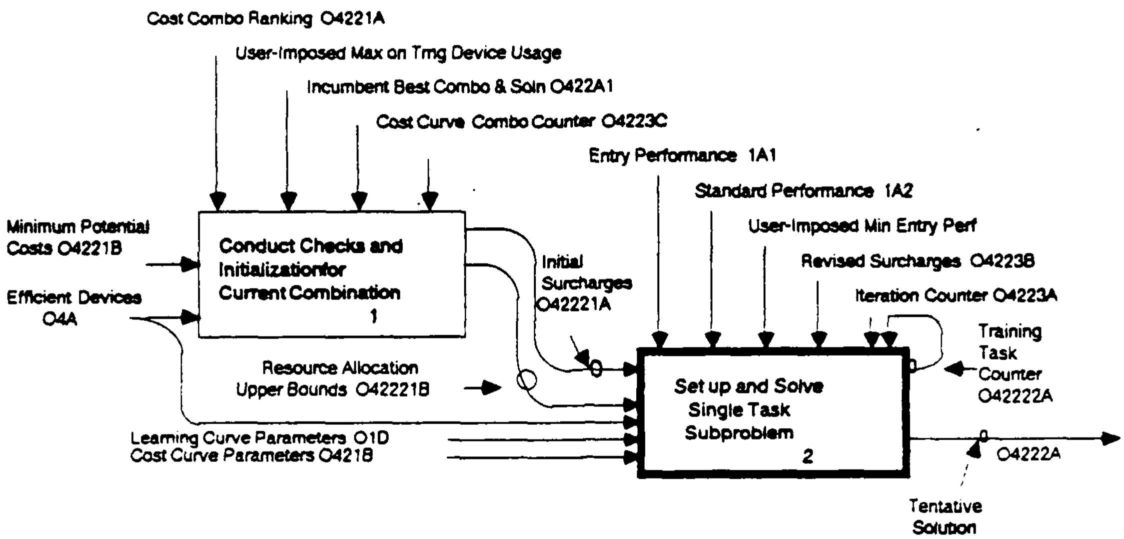
flag CONDTN set to indicate that no solution was found. If qualified devices are available, that one with the greatest $PRATE_i$ is labeled is selected as the new incumbent, I.

<u>OPT/A422223: Determine Next Switching Performance Level</u>. The next (actually, predecessor) device to be employed in the training sequence is now determined based on (1) ideal performance levels for economic transfer to other devices, (2) available resources for incumbent device, and (3) minimum entry performance constraints.

During this activity the next device, INEXT, and the transition performance level, P_{next}, are determined. P_{next} is initialized at zero, but immediately increased to P_{ENTRY}, if this is greater, and increased further to $EPMIN_I$ if this is greater. The minimum performance level, P_{MINI}, that can be reached using the incumbent device (going "backwards" down the learning curve) from P_{now} is determined based on the remaining unallocated resources, $UB_I - TRNT_I$:

$$P_{MINI} = PERF_I(-[UB_I - TRNT_I], P_{now}) .$$

If this is greater than P_{next}, then P_{next} is increased to P_{MINI}, since the incumbent device can reach no lower than P_{MINI}.



Node: OPT/A4222	Title: Determine Multi-Task Resource Allocation for Specific Problem	Date: 8/1/88	Number: OPT-35

The performance level PX_{IJ} at which the incumbent device I has the same learning rate per dollar as a challenger device J is computed in two steps, involving an intermediate variable z:

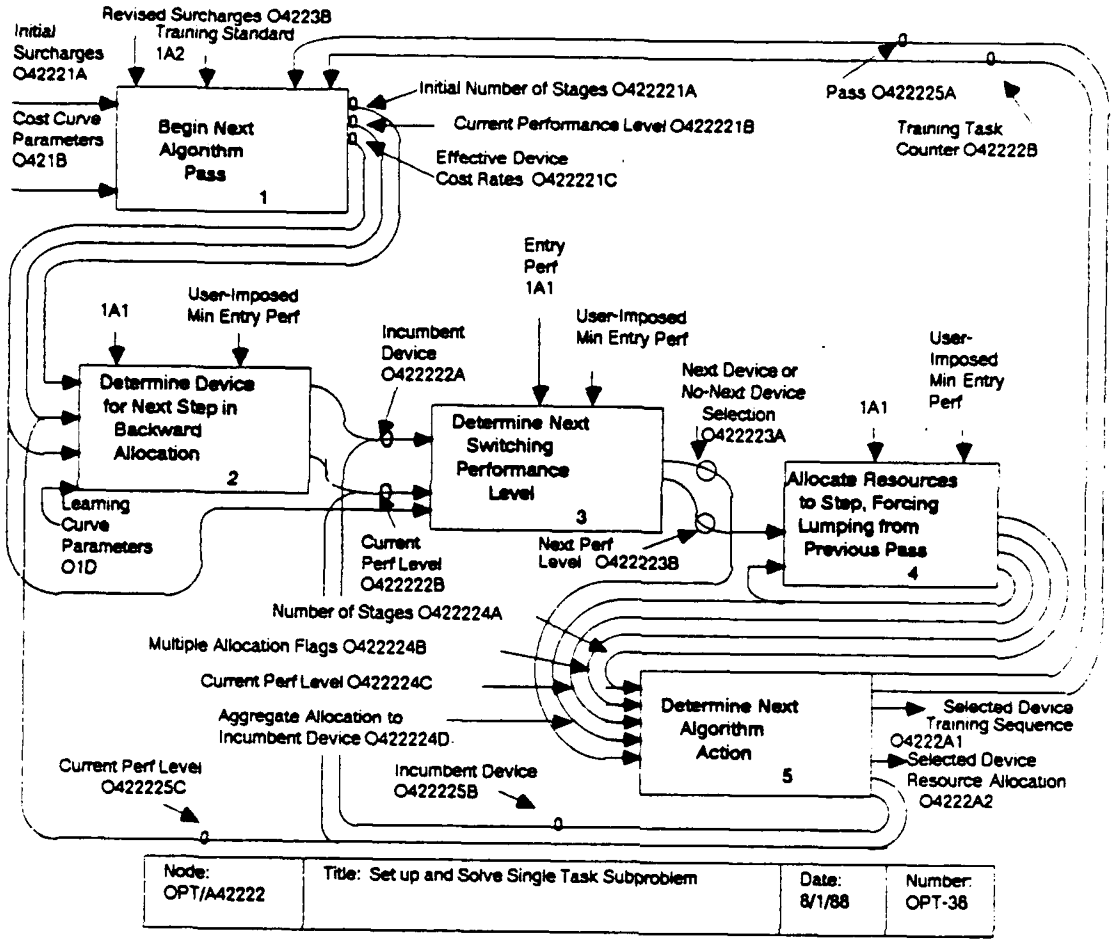
$$PX_{IJ} = (1 - z) \left[\frac{ASM_I * ASM_J}{ASM_J - ASM_I * z} \right]$$

where

$$z = \left[\frac{ASM_J * TM_J * B_I}{ASM_I * TM_I * B_J} \right] .$$

(Since PX_{IJ} will be the same at any point in the single-task resource allocation algorithm, it is only computed once, and a flag set to avoid later recomputation. Also, when PX_{IJ} is computed, PX_{JI} is then set equal to PX_{IJ} since they are equal; this obviates later calculation of PX_{JI}.)

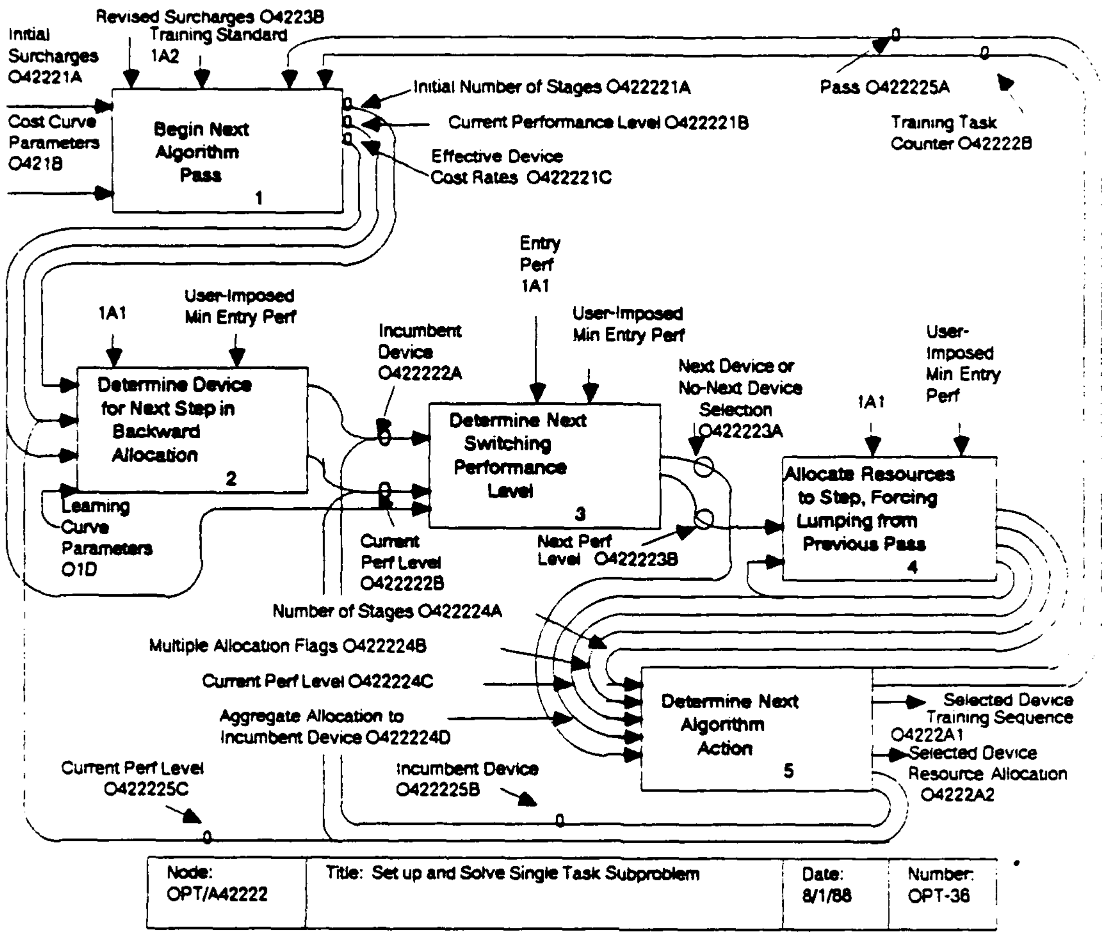
If PX_{IJ} exceeds both P_{next} and $EPMIN_J$, and is less than P_{now}, then the economic transition to J will be encountered before that to any other device considered prior to J.



Therefore, J becomes the tentative next device INEXT = J, and P_{next} is set equal to PX_{IJ}.

<u>OPT/A422224: Allocate Resources to Step, Forcing Lumping from Previous Pass</u>. At this point resources are actually allocated to the selected incumbent device, I. A counter $KALLOC_I$ is incremented to keep track of the number of times device I has received allocations on this pass. If $KALLOC_I$ becomes greater than one, then another indicator, ALLOCM, is set from "no" to "yes" to indicate that multiple allocations have occurred on this pass (so that another pass will be required to force lumping). If $KALLOC_I = 1$ and $TRNT_I > 0$, then this is the first time this device has been encountered on this pass, but it received allocation on the previous pass, possibly at multiple stages, though this information is not needed. The amount of allocation on the previous pass is needed, and is saved as $TRNT_I$. The algorithm forces these resources to be applied at this stage by (1) leaving $TRNT_I$ unchanged from the previous pass, (2) updating the current performance to $P_{now} = PERF(-TRNT_I, P_{now})$, (3) incrementing the stage counter, NSTAGE, and (4) saving the stage device assignment $IDV_{NSTAGE} = I$.

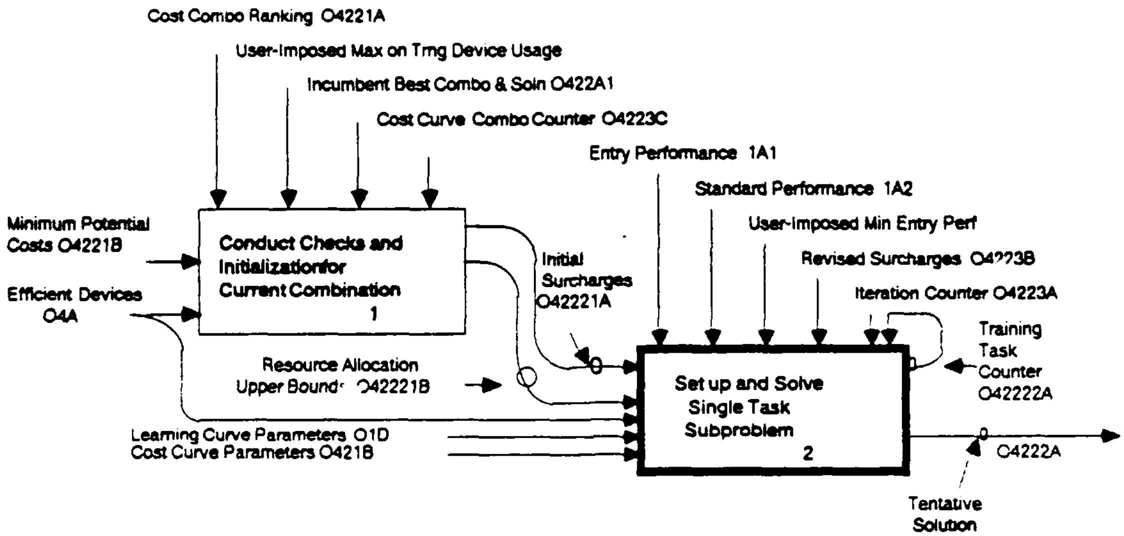
If, on the other hand, $TRNT_I = 0$ or $KALLOC_I > 1$, then either the incumbent device was not used on the last pass, or was used



earlier on the present pass. In either case, the incremental resources, TRINC, required to reach from P_{now} down to P_{next} are to allocated to the incumbent device, providing that there are sufficient resources remaining. TRINC is computed using $PINV_I(p)$, the device I task T learning curve inverse function defined in OPT/A42222: TRINC = $PINV_I(P_{now})$ - $PINV_I(P_{next})$. If $TRNT_I$ + TRINC > UB_I, then there are insufficient device I resources to allow an additional allocation of TRINC, so TRINC is corrected to TRINC = UB_I - $TRNT_I$, P_{now} is determined by P_{now} = $PERF_I$(-TRINC, P_{now}), and $TRNT_I$ is updated to $TRNT_I$ = UB_I. Otherwise, P_{now} and $TRNT_I$ are updated according to P_{now} = P_{next} and $TRNT_I$ = $TRNT_I$ + TRINC.

At the end of this activity the stage count NSTAGE is incremented and the corresponding device assignment is made: IDV_{NSTAGE} = I.

<u>OPT/A422225: Determine Next Algorithm Action</u>. If P_{now} = P_{ENTRY} (within roundoff error) then the pass is complete. Otherwise, training is still needed to bridge the gap between P_{entry} and P_{now}, so control is transferred either to OPT/A422222 or to OPT/A422223. Control is transferred to OPT/A422222 if the allocation just made was in some way constrained, and did not correspond to reaching the economic transfer level PX_{ij} for some device j; this case is flagged by INEXT having been left at an

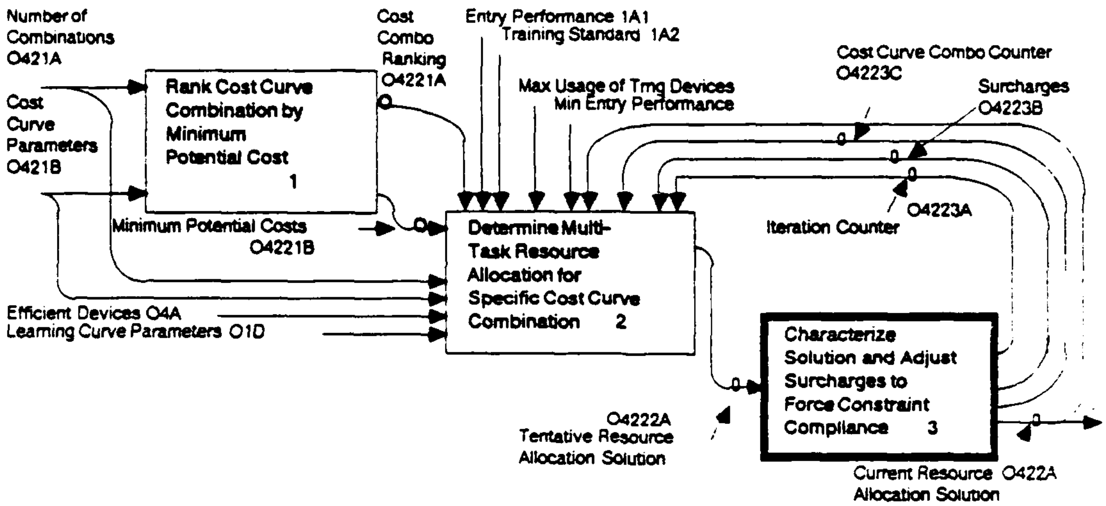


Node: OPT/A4222	Title: Determine Multi-Task Resource Allocation for Specific Problem	Date: 8/1/88	Number: OPT-35

initial value ineligible as an index of a candidate device). If the allocation did correspond to an economic transfer to an suitable next device, then control is transferred to OPT/A422223, after updating the incumbent to I = INEXT.

If the pass is complete, then the flag ALLOCM is examined to see whether multiple allocations were made to the same device. If so, then control is transferred to OPT/A422221 to begin another pass. Otherwise the solution is complete. In completing the activity some minor "housekeeping" is done, including reversing the array $IDV_1, \ldots, IDV_{NSTAGE}$, so that the numbering will correspond to training from entry performance to standard, rather than the "backward" order used in the single-task optimization.

If the number of passes becomes excessive, then the problem is deemed insolvable, and OPT/A42222 is abandoned, with the flag CONDTN set to indicate that no solution was found.



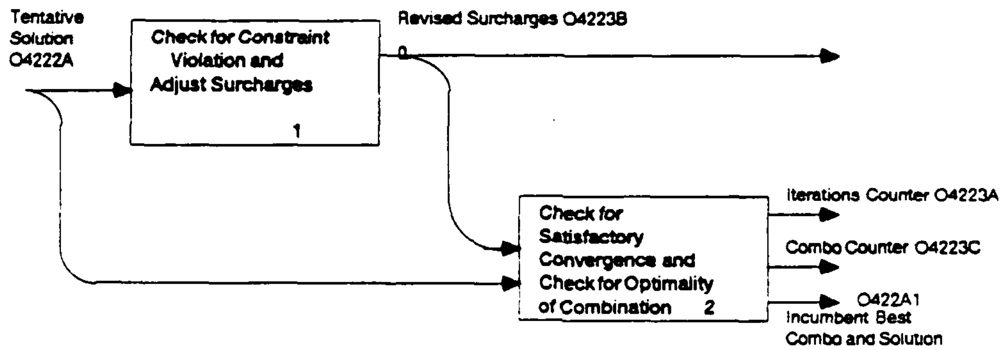
Node: OPT/A422	Title: Solve Multi-Task Resource Allocation Problem	Date: 6/1/88	Number: OPT-34

<u>OPT/A4223: Characterize Solution and Adjust Surcharges</u>.

In this activity the aggregation of single-task solutions is examined at OPT/A42231 to determine if device resource allocation constraints are respected, and whether the surcharges used to force constraint accommodation are acceptable. Also, device cost surcharges are adjusted in an attempt to force constraint accommodation and/or convergence to acceptable surcharge values in subsequent iterations. At OPT/A42232 the solution is examined in terms of surcharge values, and if the solution is acceptable (for the current cost curve combination), then it is compared against the best solution found to determine if it should become the incumbent.

<u>OPT/A42231: Check for Constraint Violation and Adjust Surcharges</u>. If $TTOT_i > TAVAIL_i$ for any device i, then the solution is deemed in violation, and a flag VIOLTN set to so indicate. If the solution is not in violation, then the surcharges in current use are saved, so that the solution can be reconstituted if needed later.

The goal of surcharge adjustment is to have $SURCHG_i > 1$ for any device i for which demand constitutes a problem with respect to usage constraints, and have $SURCHG_i = 1$ for all other devices. In this ideal situation devices are priced normally unless they



Node: OPT/A4223	Title: Characterize Solution and Adjust Surcharges	Date: 10/20/88	Number: OPT-37

are in short supply, in which case a surcharge greater than 1 is applied. The algorithm deems these constraints to be satisfied, for a feasible solution, if the surcharges and total allocations fall within a fractional tolerance of the target values. The tolerance parameter, TLRNC, is set to a small value, say 0.05.

Thus, the constraints are deemed violated if any one of the following three statements is true for any device:

$$(TTOT_i > TAVAIL_i)$$

or

$$(TTOT_i < TAVAIL_i \text{ and } SURCHG_i < 1 - TLRNC)$$

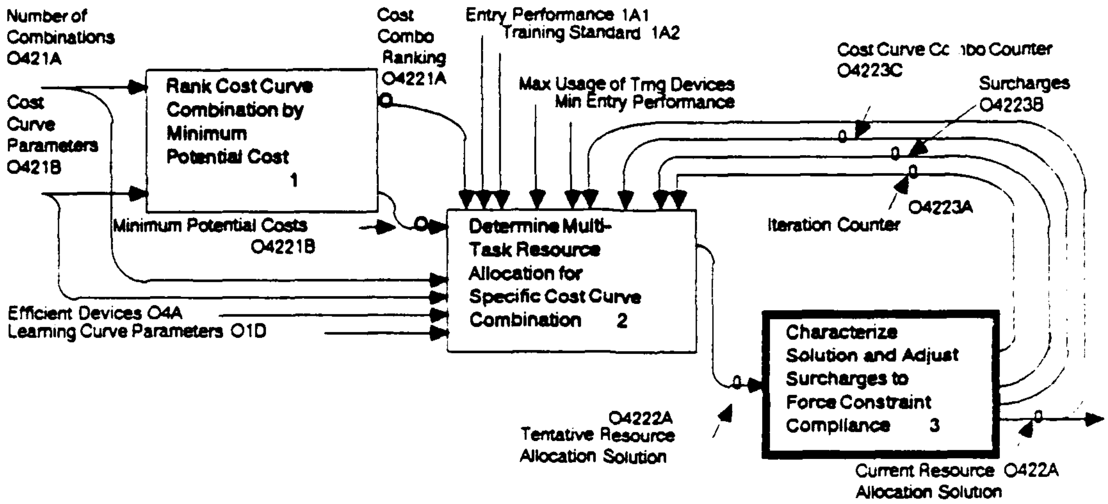
or

$$(TTOT_i < (1-TLRNC)*TAVAIL_i \text{ and } SURCHG_i > 1 + TLRNC)$$

The formulas used for surcharge adjustment are tied to three regions in $SURCHG_i \times TTOT_i$ space.

Region 1: If $TTOT_i > TAVAIL_i$, then the total training time on device i is (still) excessive, so the current surcharge is adjusted upward to

$$SURCHG_i = (SURCHG_i * TTOT_i/TAVAIL_i) * TWEAK$$



Node:	Title:	Date:	Number:
OPT/A422	Solve Multi-Task Resource Allocation Problem	8/1/88	OPT-34

where TWEAK is an acceleration parameter, set slightly greater than 1, designed to force a feasible solution ($TTOT_i < TAVAIL_i$) more quickly than would occur if the intuitively appealing parenthetical expression were used alone as the correction factor.

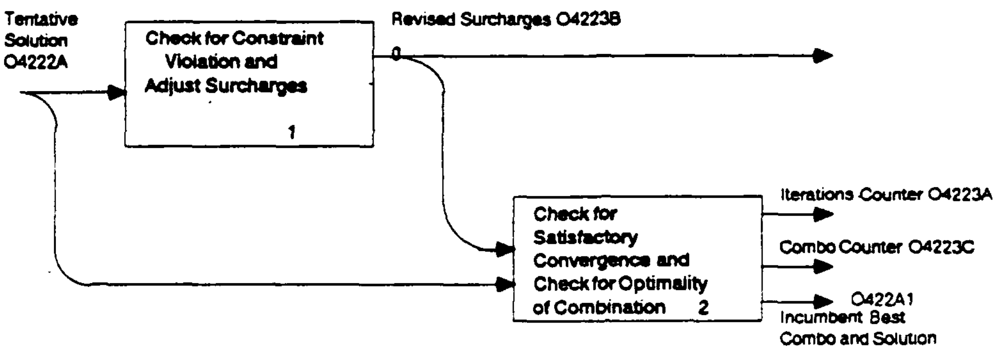
Region 2: If $TTOT_i < TAVAIL_i$ and $SURCHG_i < 1$, then the surcharge must be adjusted up toward 1. The following formula has been selected for this purpose:

$$SURCHG_i = SQRT\ (SURCHG_i).$$

Region 3: If $TTOT_i < TAVAIL_i$ and $SURCHG_i > 1$, then the surcharge must be adjusted down toward 1. The adjustment should be less drastic as $TTOT_i$ nears $TAVAIL_i$, so the following formula is employed:

$$SURGHG_i = SURCHG_i^{(TTOT_i/TAVAIL_i)}$$

The surcharge adjustments are moderated by a factor FCA (fractional change allowed). This variable is set to an initial value, FCAMAX, at the start of the examination of each cost curve combination. It is reduced by half after any iteration in which $TTOT_i$ moves from one side of $TAVAIL_i$ to the other, for some



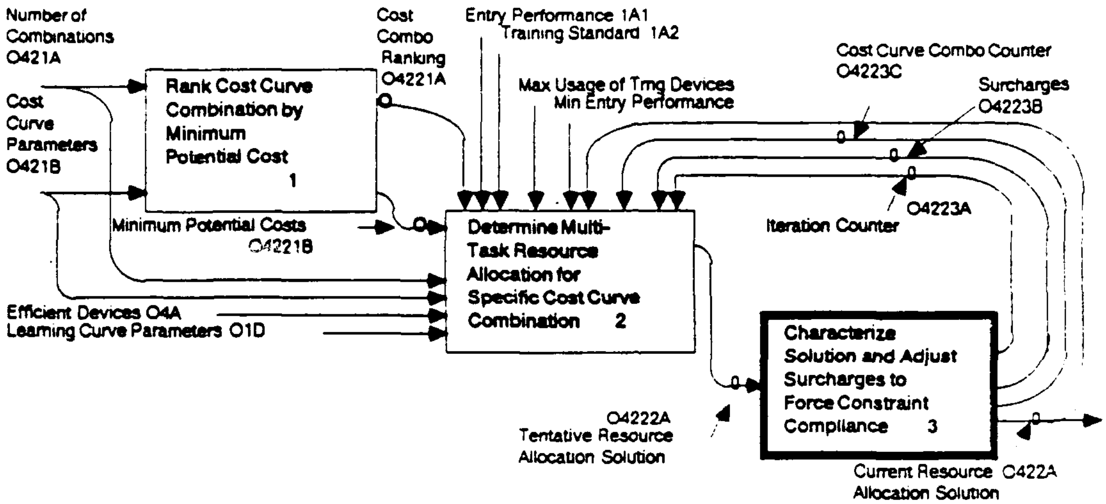
Node: OPT/A4223	Title: Characterize Solution and Adjust Surcharges	Date: 10/20/88	Number: OPT-37

device i. This effectively slows down the magnitude of $SURCHG_i$ adjustments once a solution is approached, thereby damping out excessive feedback.

OPT/A42232: Check for Satisfactory Convergence and Check for Optimality of Combination. If the solution is found to be in violation of the allocation or surcharge constraints, another iteration is begun at OPT/A42222.

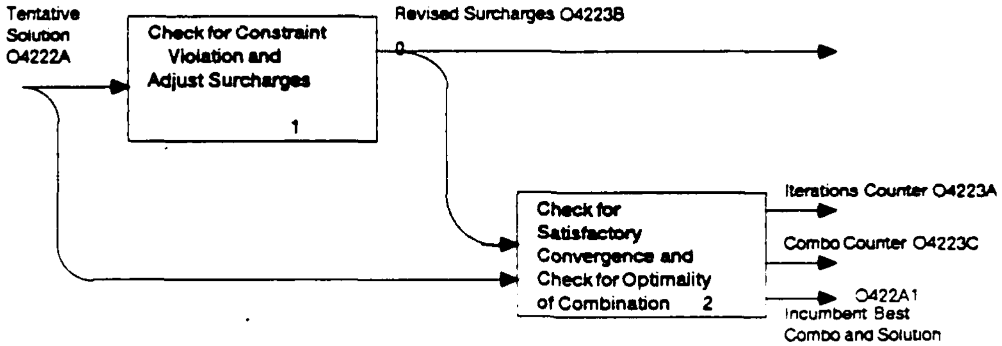
If the number of iterations becomes excessive, the iteration process is abandoned. If, in that case, the current solution is in violation of the constraints, and no feasible solution was found and saved on an earlier iteration, the current cost curve segment combination is abandoned. If the iteration process must be abandoned, but either the current solution or a past solution is feasible, then that solution is evaluated to see if it provides a lower cost overall than previous cost curve combinations.

If convergence is achieved or iteration is abandoned with a feasible solution, then the solution is evaluated using the currently detailed cost curves. If it provides a superior solution, then the current cost curve combination becomes the incumbent, KBEST = KB, and the essential solution elements are saved, i.e. the $\{TRNT_{IT}\}$, $NSTAGE_T$, and IDV_{kT}.



Node:	Title:	Date:	Number:
OPT/A422	Solve Multi-Task Resource Allocation Problem	8/1/88	OPT-34

To speed convergence and counter a tendency of the cost curves to underestimate costs when $TTOT_i$ falls outside the detailing domain, an ad hoc device is employed to give special consideration to any cost curve combination whose solution falls within the current detailing. For such a combination, the estimated cost is reduced by a factor 1/TWEAK. The true cost will be determined in the output routine for the best solution found.



Node: OPT/A4223	Title: Charactenze Solution and Adjust Surcharges	Date: 10/20/88	Number: OPT-37

EVALUATION OF THE OSBATS MODEL AND SOFTWARE

Formative evaluation is a critical component of our development of the OSBATS DSS. The evaluation effort had the following goals:

1. To determine the extent to which the OSBATS model is relevant to the training-device design problems actually faced by potential users at PM TRADE,

2. To determine the extent to which the data required by the OSBATS model are currently available, or could be made available with reasonable effort,

3. To verify that the OSBATS software accurately performs the functions specified in the model documentation,

4. To determine the ease of using the system and comprehending its outputs,

5. To determine the validity of the predictions made by the system.

Our evaluation effort has consisted of three types of activities to meet the evaluation goals. The relevance of the model, the data requirements, and the ease of use were evaluated through interviews with potential users. These interviews were first conducted when the individual DSS modules were developed. The initial interviews were quite informal, and were combined with a briefing of the function of the module. The results of the initial interviews were used to guide the development of later modules. After all modules had been developed and integrated, we conducted more extensive and formal interviews with engineers from PM TRADE. These interviews evaluated all modules in the OSBATS system. Some of the results of these interviews are being incorporated into the OSBATS software. Other results will provide the basis for recommendations for model and software enhancements.

The accuracy with which the software represents the model was verified by comparing the results of the software to selected problems with results calculated by hand. Where discrepancies were found, we determined the source of the discrepancy and communicated our results to the software-development staff. Our verification of OSBATS version 1.0 found several discrepancies. Our evaluation of OSBATS version 1.1 verified that these errors were corrected. We are now confident that the calculations used in the software correspond to the documentation of the model presented in this report.

The validity of the predictions of the OSBATS model received the least emphasis in our evaluation effort. The main procedure used to investigate model validity was to conduct several sensitivity analyses, in which model inputs were varied and the resulting recommendations inspected. These analyses were

conducted to ascertain whether changes in input values had expected results, and to discover which inputs variables were the most important in determining the model recommendations. The results of the sensitivity analyses were also used to guide the revised research plan.

A final aspect of the evaluation, the generality of the model to new training domains, was investigated by applying the OSBATS model to armor maintenance training. The details of that analysis are reported in a later section.

Evaluation of Software Modules

We conducted a formative evaluation of the individual OSBATS prototype modules (three of five) as they were delivered to ARI/PM TRADE. This formative evaluation consisted of directed interviews with PM TRADE engineers as newly delivered modules were demonstrated. The primary goal of our formative evaluation was to provide feedback from potential users to the staff members involved with ongoing development. Specifically, the evaluation of individual modules aided OSBATS development in three ways. First, information from the interview-based evaluation of modules guided modifications to the DSS interface. PM TRADE engineers readily offered opinions on the useability and comprehensibility of data organization and displays. A summary of their comments is provided below. Second, the module evaluations aided ongoing design by providing us with information about standard methods that engineers currently use in concept formulation. Third, engineers provided feedback on the correctness or appropriateness of the methods used in OSBATS. The information we have obtained in regard to the last two issues, however, has been limited. Short, interview-based evaluation sessions on separate modules do not allow PM TRADE engineers sufficient time to comprehend in detail the algorithms and general processes that underlie OSBATS.

Results of the Interview-Based Evaluations

Five OSBATS modules were delivered to ARI/PM TRADE. As mentioned above, HumRRO and ARI staff conducted interview-based evaluations of the first three with PM TRADE engineers. Typically in these sessions, one or two engineers were led through the module and instructed in its use. These sessions were structured but flexible, allowing for discussion of data selection, data displays, processing methods, and the mechanics of the interface. The data collected in these sessions served as a record and fed back into the development process. In this section, we summarize the general recommendations that address DSS design. Detailed results have been presented in a previous report (Sticha, Singer, Blacksten, Mumaw, & Buede, 1987).

The general recommendations are listed here under five categories that reflect function. These recommendations largely represent options or capabilities that the engineers would like to have added to the DSS.

1. Display format.

   a. There were concerns with the types of scales used. For example, normalized scales may read better as percentages, and ratio scales were preferred to interval scales.

   b. When costs are shown, they are usually shown as normalized cost; it may be better to provide raw cost data also. A multiplier that allows normalized costs to be converted to raw cost could be included.

   c. When total cost or total benefit are shown and package values are shown, it is helpful to show percentage cost.

   d. When developing an optimal package from the benefit-cost curve, it is useful to use a sub-optimization to locate points on the curve between optimal points.

   e. Graphs typically show benefit on vertical axis and cost on horizontal; the option to reverse the axes should be provided.

2. Sufficiency of Display:

   a. Total cost and benefit are useful in many tables.

   b. When developing optimal packages, an asterisk should be used to mark package currently being considered.

   c. When using a derived benefit score, it is helpful to have a breakdown of its meaning.

   d. Need to have global model and data settings indicated.

   e. Full name of task and its characteristics should be available (in general, a better presentation of tasks is needed).

   f. There needs to be an easy way to save a newly defined task cluster.

3. Getting Around:

   a. A standard main menu should be developed, including the options for, Task Library Menu, Examine/Modify Task Set (or Training Device Set), Display Results, Save Changes, and Quit.

   b. Consistency in menu item selection should be maintained.

   c. Consistent method for moving between levels and quitting should be developed.

d. Help screens and instructions are needed throughout.

e. There is a need for an aid to monitor one's location in the menu hierarchy.

4. Aesthetics:

a. A grid (or something similar) to aid reading tables and graphs should be added.

b. There is a need to provide "clipboard" to save results of analyses.

c. On bar graphs, vertical bars are preferred to horizontal.

d. More consistent use of colors is needed.

e. There is a need for consistent use of increment/ decrement keys.

5. Confusion Reduction:

a. The keyboard should be shut off when the mouse is being used.

<u>Use of the Evaluation Results</u>

The following guidelines, derived from feedback during the interview-based evaluation sessions, have been incorporated into the revised version of the OSBATS software.

1. A standard main menu format was adopted.

2. The capabilities of the mouse to select options and move between displays was increased.

3. The task-set-selection screen was improved. Initially, two menus were used to select a task set, requiring four key strokes. We removed one menu by structuring commands more efficiently.

4. The task-set-definition screen was improved; the mouse is used to exclude or include tasks for analysis.

5. The number of menus was reduced, and menu structure simplified.

6. The current task set and the values of other critical variables is now displayed on all results displays.

7. The ability to save model changes was improved.

8. In the Instructional Features module, normalized cost was converted to actual cost.

9. Help screens were developed for all results displays.

10. The ability to save a results screen and compare results was developed for two modules.

11. Graphical displays were redesigned, and the use of color was standardized over all model displays.

12. Longer, more meaningful names for tasks, training devices, and instructional features were used wherever possible.

Many of the other recommendations were considered and rejected, either because they involved major changes to the model, or because it was not clear to us that these recommendations offered the best solution strategy.

Structured Interviews of Potential Users

To obtain input from potential OSBATS users, we interviewed five PM TRADE engineers. Each interview was structured to include a directed demonstration of OSBATS and a set of questions about its operation. The interview addressed each of the five OSBATS modules, and each engineer evaluated (a) the presentation of data, (b) the clarity of the module's results, (c) the validity of the module's approach, (d) the availability of the data required by the module, and (e) the degree to which all relevant information was included in the module. As a group, these engineers have been involved in training-device development for an average of 12 years, with a range of 4 to 20 years. Their average time with PM TRADE has been 10 years, with a range of 2 to 20 years. The following is a summary of their comments, organized by module.

Simulation Configuration Module

Data presentation. Comments were made on the presentation of the module results and on three of the module displays. There were opposing views about whether presenting actual costs would be more beneficial than presenting normalized costs. Several engineers suggested that including both would be a good alternative. There also was concern about being able to understand how costs are normalized. (a) Results Summary Table. Only one evaluation indicated that this table was useful. Several engineers had questions about the derivation of the data in this table. (b) Task Information Box. Three of the five engineers thought this display was necessary and that it would not be possible to identify tasks without it. (c) Sensitivity Analysis Display. The engineers were not impressed with this display. One engineer said that he thought a sensitivity analysis should show what effect the cost of one item has on the cost of the whole package. Another engineer commented that "the grouping does not indicate what areas to look at for tradeoffs."

These comments indicate that the purpose of this display was not clear to the users.

Module clarity. One engineer had problems relating costs to task complexity. He commented, "Costs need to be defined in some detail and they need better explanation. All these things need examples to see what's going on." Another engineer wanted an explanation of the Cost-savings presentation. A third engineer indicated that he would like a "demo" to explain the purpose of raising the slope of the Cost-Benefit Boundary and changing the location of the Task Complexity Line. A fourth engineer asked, "Why should I care about the boundary line and how does it get established to begin with?" The most obvious points of confusion were related to the derivation of the costs and cost savings and the effects of changing the Task Cluster boundaries.

Validity of approach. The engineers were asked whether the different approaches seemed reasonable and whether they used the same approaches. One commented that he liked the general idea of indicating the complexity and cost of training tasks. With regard to developing the task sets, the evaluations indicated that the approach currently used is a task-by-task analysis. One engineer said that he does the same kind of evaluation "assuming that a functional task analysis has been done." Another engineer said he did not use that kind of evaluation but made estimates based on task analysis. In response to whether similar tradeoffs are conducted now, a third engineer said, "Tradeoffs are done in terms of importance of tasks--one versus the other." However, he did indicate that the OSBATS tradeoffs seem reasonable. In relation to the cost axis, an engineer remarked, "This relates to what I do in intuition when looking for high cost-driver tasks." In general, the engineers thought the approach and algorithms used in this module were reasonable. Their major concern seemed to be the accuracy of the data the module uses. One engineer remarked that the validity of the approach to cost and simulation "becomes an issue of data acceptance."

Data availability. One response to the question of the availability of data was that the information is there but that it is TRADOC information and PM TRADE does not normally receive it. Another response was that the information exists but the school hides it. This engineer said that the school is required to provide the information but hides it because "they are not sure the data will get them what they want." The remaining three engineers answered that it would be a "herculean effort" to obtain these data. When asked how similar this information is to what is normally available, one engineer responded, "Nothing like this is available. This is the kind of info needed and what I'd like to get but don't usually have."

Other relevant information. One engineer thought there should be additional categories. He remarked that there is a "need to talk about specific subsystems; a weapons system trainer isn't a full mission simulator but is more than a part-task or mission subset." Only one engineer suggested a need for other task

cluster information. He wanted "mission essential information." In general, the engineers were satisfied with the task cluster information provided by this module.

<u>Instructional-Feature Selection Module</u>

<u>Data presentation</u>. There were few comments on data presentation in this module. One comment was that the only columns necessary on the Results Display are the Benefit, Cost and Ratio columns. Another engineer also felt that the "percentages are hard to understand." This engineer was referring to the normalized benefits and costs that he interpreted as percents. This indicates that additional explanation of normalization is needed for the user to understand the results.

<u>Module clarity</u>. The engineers' comments showed that the information presented in this module is not easily understood. Two engineers said they understood the information though they still wanted access to an explanation, especially on the inclusion and exclusion of weights. The remaining three engineers said they did not understand the information presented. One engineer commented that he would like to have an explanation of both the derivation of benefit weights and the normalization scales used. Another engineer commented, "The benefits are less defined for me here," indicating that the derivation of the benefits is not clear.

<u>Validity of approach</u>. All five engineers agreed the instructional feature cost-benefit tradeoff is necessary for concept formulation. When asked if the analysis performed in this module is similar to tradeoffs normally performed, the answers varied. One engineer answered that he tries to get TRADOC to assign the benefit values while he provides the costs and does the analyses of cost drivers. Another engineer said that he does it task by task, figuring how many tasks are trained with a feature versus the cost of the feature. This engineer must not have been aware that the instructional feature benefits are calculated in a similar manner in OSBATS.

<u>Data availability</u>. Two of the engineers indicated that the data are available but are not compiled and analyzed the way they are presented in OSBATS. One engineer responded that the information exists but is not easily available. The remaining two engineers, who happened to be the engineers with the most experience in training-device development, indicated that they did not think the data exist. This suggests that gathering the data necessary to support this module may be difficult.

<u>Other relevant information</u>. Though two engineers indicated that training-device development engineers should be familiar with state of the art instructional features, other engineers responded that they would like more information about the instructional features on the Instructional Feature Selection screen. One engineer asked for more information about the

various instructional features because their names were not sufficiently descriptive. Another engineer wanted a list, by trainer, of the general categories of instructional features that apply and felt the user should be able to add to this list. In addition, two engineers mentioned that the school sometimes indicates which instructional features need to be included in the trainer. In this case the user might want to be able to enter the school requirements into the system as the base package.

Fidelity Optimization Module

Data presentation. Three displays were addressed: (a) One engineer was curious why the user-defined package feature could not be expressed in terms of descriptions or benefits. (b) One engineer said he would like to see the change in benefit and cost for each step when incrementing the Package Display. (c) Several of the engineers seemed in favor of a comparison screen.

Module clarity. The engineers were asked if it is easy to understand what the benefit numbers mean. Only two of the engineers answered yes, and one of them said he understood only after receiving an explanation. The remaining three engineers answered no. One engineer suggested incorporating an on-line explanation on the Constrain Fidelity Dimension screen that shows where the benefits come from. Problems with the Package Display screen seem to stem from the data values rather than the capabilities of OSBATS. For example, one engineer commented that having the increment/decrement feature to clarify the order of selected packages "is wonderful if you believe these data. Problem is having to tailor the data for each problem set." Another engineer said of the same feature, "I'm sure it will help you on your misguided way. The reason I say that is that I don't agree with your benefit and cost." The engineers seemed to have the most difficulty with comprehending what the benefit numbers meant, and with accepting the module data.

Validity of approach. Three engineers had problems with the rationale behind the determination of benefit for the fidelity levels. One engineer commented, "My problem is that you are sometimes trying to see something that's smaller than can be displayed. I may not be interested in this question because state of the art can give resolution at 2 km, so 300 meters is irrelevant." Another engineer remarked that he had some problem with the data also. Though most of the engineers did not understand the benefit numbers, they indicated that they understood the rules and algorithms used in this module. For example, one engineer responded, "Your explanation is easy to understand but not from using the system." It is not clear how he came to understand these calculations.

Data availability. Each engineer was asked if the data or information required by this module is normally provided by the school. Three of the engineers answered no. One engineer said that the school does not provide the "fidelity stuff." The other engineer said the school is weak with cost data but "provides

fidelity and task characteristics within reason." These responses indicate that the schools either do not provide this information or are inconsistent in providing it to the development engineers.

Other relevant information. The major concerns in this category were the input and output of the system. The engineers were asked whether the set of fidelity dimensions and levels used in this module is comprehensive. Two of the engineers did not feel qualified to answer. The remaining three engineers agreed that the list is not comprehensive for general pilot tasks. For example, one engineer said it is not clear what the list is addressing. He said, "It covers general categories, but you don't have enough information to tell what the picture under consideration is." With regard to output information, one engineer wanted the module to provide information on subsystem designs. Also, incorporating school requirements such as schedules into the system is important to the engineers. Another engineer suggested a time-line of training-device development because certain fidelity dimensions take longer to develop and time is a realistic constraint. Finally, two engineers mentioned wanting to see the "training effect" or "flow diagram of impact on system."

Training-Device Selection Module

Data presentation. Engineers were asked to evaluate the value of the graphs in this module. The first graph presented, called Graph This, shows the data presented in the Results Summary Table. Three of the engineers thought the graph was better for concept formulation documentation than the table. The only concern was that the graph is more difficult to reproduce. Tables are easier to print. The Individual Task graphs were considered less useful. Three of the engineers either did not understand the purpose of the graph data or did not see a need for it.

Module clarity. The areas of confusion in this module were the Individual Task tables and graphs. One engineer remarked about a task graph, "What is this showing me? It doesn't look worth it to use the device for such a short time. It would be more effective to use the actual equipment. This doesn't give me an increase in benefit over the actual equipment." Another engineer also said that the task graph indicated to him that actual equipment "is best." The engineers seemed to interpret the task graph as a graph comparing devices rather than showing the effects of a training system. More explanation is probably necessary for the Individual Task table and graph displays.

Validity of approach. Four of the engineers said that the analysis done in this module is similar to what they do. There were a few suggestions, however, worth noting. One engineer remarked, "This only addresses inventory that can address part of tasks. What should be addressed is whether the tasks not addressed by existing trainers are worth the cost of developing a

new device, like the Hardman." Another engineer said he would like an "interactive dialogue," where the user enters system requirements and the display provides the design features necessary to get the desired results in training.

Data availability. Two of the engineers thought the data required by this module would be very difficult to get. Two other engineers thought the data are generally available, but they would have problems gathering and organizing it. Several of the engineers said they have this kind of data available to them during the concept formulation process.

Other relevant information. One engineer commented that he considers the number of students to be related to the student-to-instructor ratio, which is not a factor in OSBATS. Two engineers remarked that the life cycle is usually considered to be 20 years rather than 10 years. When asked about additional output, several engineers wanted time schedules for development and delivery.

Resource Allocation Module

Data presentation. The engineers were asked what they thought about the accessibility of the screens in this module. The responses were negative. One suggestion was for a pull-down menu. Another engineer said he likes the Macintosh computer and its features. A third engineer simply responded that the organization could be optimized.

Module clarity. The results of this module are presented in a format similar to the format of the Training-Device Selection Module. Comments about the tables and graphs made in that module apply to both modules.

Validity of approach. Four of the five engineers said that the analyses performed in the Resource Allocation Module are similar to the analyses they perform. One engineer stressed that he always does several iterations when designing a training device. He may not have realized that OSBATS can be used in an iterative fashion.

Data availability. The engineers indicated that the number of students required in training per year is always provided by the schools. With respect to the availability of the data necessary for the module, two of the engineers felt the data would not be available. The remaining three engineers said "sometimes," "normally" and "usually" the data is available.

Other relevant information. A number of constraints were mentioned that were not incorporated into OSBATS: (a) size of the door in the hangar, (b) transportability of the device, (c) power cycle, (d) cost schedules, (e) school-imposed constraints, (f) student-to-instructor ratio, (g) floor space, (h) physical building changes, and (i) operational hours.

General System Features

The engineers were asked to evaluate the Help and User Comment system features. Each evaluation contained specific ideas about what a help screen should include. One engineer said that he would like an "index of help," with the details of the cost elements and Indication for Simulation explained. Another indicated that the help function "should come as close to a turnkey operation as possible -- completely explain functions and options." There was a suggestion that the help screen should allow selection of the level of instructional detail the user desires. There was also a comment that the screens should include larger type, color graphics and less text. Four of the five engineers thought the User Comment feature was a useful feature. However, one of the four commented that not having the screen to look at as a reference when making comments hinders good note-making.

Comments pertaining to general aesthetics of the system were also addressed. They included a suggestion to increase the size of the menu bars so that there is space around each label. There were also comments such as "problems with the arrow," "need a brighter mouse" and "feedback on mouse" from three different engineers, which seems to indicate that there was difficulty keeping track of and controlling the cursor.

Summary

Many of the comments made by the engineers represent general areas of concern that should be considered in further OSBATS development. Descriptions of these general areas follow:

1. OSBATS does not accommodate school requirements as constraints in the training-device development process. For instance, in the Instructional Features module, several engineers mentioned that the school often specifies certain requirements. Levels of instructional feature packages might be more beneficial if the user could enter into the system the features that are specified by the school and should be included in the base package.

2. A major concern is the cost data and the cost-savings associated with simulators. Engineers like cost comparison charts. Several engineers indicated that training-device design is a process driven by cost, time schedules and school requirements.

3. Development time seems to be an important issue that is not addressed in OSBATS. Time schedules are often constraints in the current training-device development process.

4. Engineers are confused by the derivation of the benefits, both instructional feature benefits and fidelity dimension benefits. More explanation is necessary.

5. There is some confusion about how the results are normalized. Normalized numbers are confused with percentages.

6. "Functional groups" of tasks are often used in the current training-device design process (i.e., tasks for motion systems). There seems to be an inclination to look for "grouping" and to design device components by function.

7. Several engineers indicated that training-device design usually involves concentrating on one device as opposed to a "minimum family of devices." One engineer said, "usually the school won't consider families." He said later, "Our requirements are in terms of a device, not a combination of devices." Another engineer said that he usually tries to come up with "a single design."

8. There was concern with the availability of the data required by OSBATS and the cost involved with having to gather data for each different problem addressed. One engineer asked, "How usable is the system?" His opinion is that "in its present form, not being flexible and needing lots of data," the system is limited in use. Another engineer said that it is important that the user be able to get into the system and "inspect or change data for benefits and cost."

Verification of OSBATS Software

The verification of the OSBATS software took the form of predict and verify -- prediction by hand simulation and verification by comparison to the OSBATS output. More specifically, we altered values in the OSBATS input files, hand-calculated OSBATS outputs using the correct formula, and then ran OSBATS with the altered input files to compare the program's results. This was done for each output in each OSBATS module. Verification was conducted for both version 1.0, which was delivered in July 1987, and version 1.1, which was delivered in March 1988. The results of these evaluations are briefly described in the following subsections.

Verification of Version 1.0

The verification process uncovered several inconsistencies between the model and the version 1.0 software. Some of the problems came from the fact that the model was being modified at the same time that software was being developed. Thus, for example, the Simulation Configuration Modules was using an earlier version of the learning curve than was represented in the model description, and implemented in the other software modules. Since individual modules were developed to operate independently, they were occasionally inconsistent in the methods used. The integration of the modules eliminated these inconsistencies. Finally, some inconsistencies were errors that were not discovered when the modules were first developed.

The results of the verification of version 1.0 provided feedback to the software development staff. All of the errors in discovered in version 1.0 were addressed in the development of version 1.1.

Verification of Version 1.1

Verification of version 1.1 of the OSBATS software was conducted using the same general procedure that was used for version 1.0. All aspects of the program were examined. However, special attention was paid to those areas that were newly developed in version 1.1, or in which errors were found in the earlier version of the software. In general, the evaluation determined that the methods used by the software are correct. An overview of the results is presented, by module, below.

Simulation Configuration. Our evaluation of the Simulation Configuration Module verified that the errors of the previous version were corrected, and uncovered some other minor discrepancies.

1. We tested this module to verify that the new cost curve formulas had been incorporated into this version. We carried out this testing by comparing Simulation Configuration results with Training Device Selection results when the Training Device Selection Module was run with the visual meteorological conditions (VMC) trainer and the actual equipment at costs equal to the variable hourly cost. For the tasks tested (chosen randomly), the value of Device Hours Used calculated in the Simulation Configuration was comparable to the time allocated to the VMC trainer in Training Device Selection.

2. The system calculates the Cost Savings only once. If the task cluster is changed during a run of the program, the cost savings figures from the previously used task cluster are retained.

3. From the Individual Task display, the Task Complexity Line can not be adjusted. Selecting Task Complexity Line causes the line to become 0.1 and remain there.

Instructional-Feature Selection. This version contains the newly-incorporated Instructional Feature Rule Base. We verified that the rule base is printing its output file correctly. This involved running the rule base for a random ten of the 38 tasks and verifying that the results of the rule base were reflected accurately in the output file.

Fidelity Optimization. This version of OSBATS contains a Fidelity Rule Base which generates the a matrix of cue and response requirements by task. In addition, the task economic weights are now calculated by the system, rather than being read directly from a file. To verify that the benefit and cost of each level of each fidelity dimension is being calculated correctly, we chose one dimension, Visual Resolution, and hand

calculated the benefits and costs across all levels of this dimension. We found that the task economic weights, fidelity benefits, fidelity costs, percent benefits and percent costs are being calculated correctly.

Training-Device Selection. The calculations in the Training-Device Selection module have changed little since the last version. However, re-verifying these calculations enabled us to verify other calculations performed by the system.

1. We chose a single task, Task 1054, and hand-calculated the slope of the learning curves, the crossover points from training on one device to another, the time and cost of training on each training device and the performance levels achieved on each device. Our results were identical to the program results. This indicates that the device asymptotes and time multipliers are being calculated correctly.

2. We then went through the calculations again to verify the Re-iteration function, which performs the calculations using the calculated annual utilization ($UACT_i$) for each task in place of the given nominal utilization ($UNOM_i$). The results of out hand calculations were identical to the program results.

Resource Allocation. The Resource Allocation Module is the most complex of the modules, and was revised more extensively than the other modules. Our evaluation of this model found several problems with both the model calculations and the user interface.

1. We found that the Resource Allocation optimization calculations do not necessarily find the least expensive solution.

2. When selecting Default from the "Select Default/Modified Constraints" window, the Default constraints are those constraints specified most recently. To view the results with no constraints after constraints have been used, the user must return to Select Problem and, in effect, restart the module.

3. The name of user-designed training devices are not always represented correctly by this module. On the Show/Edit Constraints and Individual Task Graph displays, a device named "sevenmillion" might show up as "to".

Conclusions. The evaluation effort found that, in general, the software was making the calculation correctly, in correspondence with the model documentation. Several discrepancies were found, however. Most of them were minor, and all have been corrected. We are currently confident that the calculations performed by the software correspond to the model description presented in this report.

Sensitivity Analyses

Sensitivity analysis refers to a set of procedures used to assess the effects of manipulations of a model's variables on the results of the model. These techniques rely on making systematic changes in the values of input data and measuring the resulting changes in model results. The sensitivity analyses had the following two goals.

1. To determine the responsiveness of the model to variable manipulations, and

2. To ascertain the validity of critical model assumptions.

Because sensitivity analyses do not involve empirical human performance data, they do not provide a rigorous test of the model validity. Rather, they test validity by ascertaining the correspondence of model predictions to general trends and expectations. We describe the results in relationship to these two goals.

Model Responsiveness

We addressed model responsiveness by varying the major model inputs over a wide range of values and determining the effect on the model recommendations. In most cases, we used the results of the Training Device Selection Module as the main dependent measure to assess sensitivity. The following subsections summarize the results of the sensitivity analyses.

Task training requirements. We investigated the effects of changes in entry performance level, performance standard, and task cue and response requirements. Task cue and response requirements are calculated by the fidelity rule base, rather than basic input data. However, the task cue and response requirements were used as the basis of the sensitivity analyses, rather than the task activity data on which they are based, because there are fewer cue and response dimensions than task activity dimensions, and because the task cue and response dimensions are numeric, and more amenable to sensitivity analyses.

We found that lowering the entry performance causes the less expensive devices to be used in the initial training of a task. However, the performance at which training switches from the less expensive device to the more expensive device (the crossover point) remains the same. The entry performance affects the training times through the head start and the time scaling factor.

Increasing the performance standard causes a roughly proportional increase in the crossover points. Thus, when the performance standard is increased, the same training devices are generally recommended. This somewhat counter-intuitive result occurs because the performance standard is used in the definition

of the cue and response requirements. That is, the cue and response requirements are defined as the fidelity required to reach the performance standard. Thus, changing the performance standard without making the corresponding change in the cue and response requirements would have little effect on the results. The performance standard affects the training allocation through the benchmark asymptote and the time scaling factor.

Task cue and response requirements are among the most critical variables in determining the recommendations of the OSBATS model. Increasing the cue and response requirements causes a decrease in the device asymptotes. For example, for a task with many high fidelity requirements, increasing the Audio requirement from 0.54 to 0.90 caused the asymptote to decrease roughly five percent. For a low-fidelity task, increasing the Audio requirement from 0.54 to 0.90 caused the asymptote to decrease roughly 15 percent. An increase in the asymptotes causes a shift in the training allocation away from low-cost devices toward the sophisticated training devices and actual equipment.

Other Task Data. The major variables that were investigated here involve estimates of training hours and costs. We found that multiplying through the task training hours (2A1 - 2A5) by a constant factor produces a change in training times by exactly that factor. The relative proportion of training time allocated to each device does not change, however. This result indicates that the devices chosen by the Training-Device Selection Module are completely insensitive to the task training hours, when all five of the data elements are varied in proportion. Changes in training hours would have an impact on the results of the Instructional-Feature Selection and Fidelity Optimization Modules, however, because they would affect the task weights.

We found that increasing the setup hours by a factor of ten without changing other task training times causes a marked increase in training times with respect to both the training devices and the actual equipment. In the tasks we examined, an increase by a factor of ten caused the training time to double.

Training Device Data. The training device cue and response capability ($FCAP_{ij}$) has a large effect on the model recommendations that is analogous to the task cue and response requirements. The effects are not completely analogous, however, because the effect of $FCAP_{ij}$ is moderated by the minimum performance parameter, $FMIN_j$. We found that the change in $FCAP_{ij}$ has the largest influence on the asymptotes when the minimum performance parameter is low. We set the $FCAP_{ij}$ for all dimensions equal to 1.0 except for one dimension we call j. We varied the $FCAP_{ij}$ for this dimension from 0.0 to 1.0. At $FCAP_{ij}$ = 0.0, the task-device asymptote is $FMIN_j$. At $FCAP_{ij}$ = 1.0, the asymptote is 1.0. As $FMIN_j$ increases from 0.0 to 1.0, the asymptotes fall increasingly close together as $FCAP_{ij}$ is varied. Thus, the largest changes occur at the lowest values of $FCAP_{ij}$ when $FMIN_j$ is also low.

Fidelity Dimension Data. We examined two fidelity dimension data variables, fidelity dimension cost and the fidelity dimension minimum performance parameters.

We found that as the exponent in the fidelity dimension cost estimating function gets very large (ten times its original value), the cost curve becomes increasingly less linear. Low levels of technology are still cheap. Higher levels of technology get dramatically more expensive.

Lowering the minimum performance parameters for all 11 fidelity dimensions to zero causes a decrease in training on the relatively unsophisticated devices such as the NOE trainer. Increasing the $FMIN_j$ for all eleven dimensions to 0.90 causes a shift in allocation of training towards the Weapons and NOE trainers and away from the VMC trainer. We concluded that increasing the FMINs benefits the low-cost trainers at a greater rate than the high-cost trainers.

Training System Data. We examined three training system variables, the maximum instructional feature effect, the assumed setup savings percentage and the maximum number of instructional features.

We feel reasonably confident that SUSAV, the setup savings percentage, is greater than or equal to 50% but less than or equal to 80%. We also feel confident that IFBN, the instructional feature benefit, has a value greater than or equal to 0% but less than or equal to 50%. Therefore, the time multiplier varies from 1.0 to 2.73 as IFBN and SUSAV vary. At a glance it would seem that this fuzziness would have enormous implications on the learning curve. However, it is only in the first few hours of training that the time multiplier affects the learning curve more than the asymptote.

We feel reasonably confident that the MAXIF, the maximum number of instructional features that benefit training, is between 2 and 5. Thus, the largest amount of fuzziness in the time multiplier is approximately 0.06' comparatively little. Therefore, the fuzziness associated with the value of MAXIF has very little affect on the OSBATS model.

Implications of Analyses on Validity

Several of the results of the sensitivity analyses help confirm the validity of the OSBATS model, in that they correspond to our expectations. Other results offer important characterizations of the model that would provide the basis for a critical test of model assumptions. The following are some of the more important results, restated in somewhat more general terms.

1. Simpler training devices are more appropriate at lower skill levels; higher fidelity training devices are more appropriate at higher skill levels.

2. The most critical process in the model application is the process that determines task cue and response requirements.

3. The model predicts that fidelity can be sacrificed more readily on less critical fidelity dimensions (with higher $FMIN_j$) than it can on more critical fidelity dimensions.

4. Fidelity is more important than instructional features at high skill levels; instructional features are more important at low skill levels, although it is difficult to say at what skill level fidelity becomes more important than instructional features.

5. Fidelity requirements are related to the performance standard. If the standard is raised, then it may require greater fidelity to train to that standard.

6. The total training hours required to train a task do not effect which training devices are selected for that task, but it does affect the total cost of training.

Most of these results are expected, and serve to give us some confidence about the validity of the OSBATS model, albeit on a very informal level. Other results, particularly those that relate the importance of fidelity and instructional features and tie fidelity requirements to the performance standard, provide the opportunity for critical tests of the OSBATS model.

EXAMPLE APPLICATION OF OSBATS

The OSBATS system is composed of five modules developed in the C programming language, and two rule bases developed using EXSYS (Copyright Exsys, Inc.), an expert system shell. Installation of OSBATS requires an IBM PC/AT, Zenith 248 or compatible with 640K of memory and a 10 megabyte hard disk. Additional hardware requirements include an enhanced graphics adapter (EGA) and color monitor, an 80287 numeric coprocessor, and a Microsoft-compatible mouse.

The user-interface consists of a variety of menus and displays. After the software is installed and OSBATS has been started, the user interacts with the software using the mouse. The various modules can be accessed through the Module Menu. Likewise, each module has a menu of its own outlining its subsections. Each module subsection contains a display or series of displays. A display may be a graph or a table of results or it may be for the selection of constraints or for saving results. The user can access various displays within a module subsection through a menu at the bottom of each display. OSBATS also includes a HELP feature to assist the user, which can be accessed from any display. The figures presented in this section resemble the displays produced by OSBATS, except that the menus are omitted.

This section describes an application of the OSBATS model to an artificial, but realistic problem related to Army Aviation training. The goal of the description is to illustrate both the kinds of solutions to concept-formulation problems provided by the OSBATS model and the ways that the individual modules interact to produce these solutions. The example should provide contextual information to help the reader understand the specific calculations presented in the formal model description. This section will not describe the calculations in any detail. Rather, it will concentrate on how the results of different calculations can be integrated to provide overall support to the concept-formulation process. The results presented in this section were calculated by the decision support software that implements the OSBATS model. User and programmer documentation for this software are presented in separate reports.

The sample problem is based on an existing training system, the AH-1 Airman Qualification Course (AQC). The section begins with brief descriptions of the training requirements for the course, and the data collection methods. The section continues with an overview of the sample analysis. The OSBATS model is a flexible set of tools that can be tailored to the needs of the problem and the preferences of the user. In the example problem we have selected a minimal analysis that illustrates the use of each module. Following the analysis overview, we discuss the operation of each of the modules. Finally, we summarize the results of the analysis and describe their implications for this hypothetical problem.

Description of the Example Problem

To exercise the models developed during this project, and to evaluate the model's data requirements, it was necessary to develop a training data base for a device-based training system. We chose the AH-1 AQC to form the basis of the initial model application.

The AH-1 AQC is a training course designed to qualify aviators to fly the AH-1 helicopter--one of the two attack helicopters presently in the Army's aircraft inventory. The AH-1 AQC is an Institutional training course taught only at the U.S. Army Aviation Center, Fort Rucker, Alabama. During the 6.4 weeks normally required to complete the course, students receive 33 hours of flight instruction in the aircraft, 3 hours of instruction in the AH-1 Flight and Weapons Simulator (AH1FWS), and academic instruction on aircraft systems, emergency procedures, and tactics. Although some Army aviators enter the AH-1 AQC immediately after completing Initial Entry Rotary Wing (IERW) training, it is more common for aviators to enter AQC training after completing one or more flying duty assignments in another aircraft. In either case, all aviators will have been trained and become qualified in two or more other aircraft prior to entering the AH-1 AQC.

Overview of Data Collection

The procedures used to develop and collect the data on the AH-1 AQC were described in detail by Sticha, Blacksten, Buede, and Cross (1986). In general, the categories of data that were collected met the requirements of the OSBATS model, and included the following classes of information:

1. a comprehensive list of AH-1 AQC training tasks,

2. estimates of AQC students' pre- and post-training skill level on each training task (data base category 1A),

3. ratings of the extent to which safety and training efficiency considerations would favor training in a simulated environment (category 1B),

4. estimates of the amount of time and number of practice iterations required to master each training task in the aircraft (category 2A),

5. ratings of the extent to which several specific information-processing activities are required to perform each task (category 2B), and

6. assessments by task of the activities that require high fidelity in training-device components to achieve effective training (category 2C),

7. definition of a family of generic training devices and judgments about the AH-1 AQC tasks that can be trained in each device (category 3).

The data that are used in this analysis are shown in Appendix A, with asterisks indicating the tasks used in the example problem. The following paragraphs provide a brief discussion of the tasks and training devices that were used in the analysis.

AH-1 Training Tasks

The procedure used to generate tasks resulted in a list of 188 individual tasks that AH-1 aviators must learn to perform. This list includes both tasks that are trained during the AH-1 AQC (Institutional training) and tasks that are trained only after an AH-1 aviator has been assigned to an operational field unit (Continuation training). Of the 125 tasks that are trained in the AH-1 AQC we selected 38 tasks to form the basis of the example problem. These tasks were chosen to satisfy the following conditions: (a) all tasks were actually trained in the AQC, rather than in continuation training, (b) tasks represented a variety of skills, such as normal flight, malfunction procedures, and attack mission tasks, (c) tasks appeared to have a variety of cue and response requirements, from low to high fidelity, and (d) tasks could not be trained to standard using academic training only. The tasks that were chosen are enumerated in Table 7. The table lists the number of the task in the Aircrew Training Manual (ATM), the task name, and an abbreviated name used to refer to the task later in the analysis.

Family of Training Devices

One of the primary functions of the OSBATS model is to aid in defining the most cost-effective set of training devices for a training system. To evaluate the models' utility for this purpose, it was necessary to define a family of potential training devices that covers a wide range of cost and training functions and to make a subjective assessment of the tasks that can be trained in each of the devices. Working in concert, the SMEs developed descriptions of five training devices that they judged to have potential value for training aviators attending the AH-1 AQC: a Cockpit Procedural Trainer, a Doppler Navigation Trainer, an Instrument (IMC) Flight Trainer, a Weapon Systems Trainer, and a Flight and Weapons Simulator. The Flight and Weapons Simulator is the only one of the five devices that has been developed for the AH-1 aircraft, although similar devices have been developed for other types of rotary wing aircraft. The assumed design of each training device was specified by developing tables that show:

1. the type of cockpit displays,
2. the type of engine instruments (if any),
3. the type of cockpit controls,

Table 7 Tasks Selected for Example Problem

Task Num.	Task Name Abbreviation	Task Name
1003	T/O_CHECKS	Start, Runup, Takeoff Checks
1004	HOVER_PWR_CK	Hover Power Checks
1005	HOVER_FLT	Hovering Flight
1006	NORMAL_T/O	Normal Takeoff
1007	S/MAX_P_T/O	Maximum Performance Takeoff
1008	DECEL_ACCEL	Deceleration/Acceleration
1009	TRFC_PTRN	Traffic Pattern Flight
1012	DOPPLER_NAV	Doppler Navigation
1013	PRE_LNDG_CK	Before Landing Checks
1014	VMC_APPROACH	VMC Approach
1015	SHLW_APPR	Shallow Approach/Landing
1016	CONFIND_OPN	Confined Area Operations
1018	TERR_FLT_T/O	Terrain Flight Takeoff
1019	TERRAIN_FLT	Terrain Flight
1021	NOE_DECEL	Nap-of-the-Earth Deceleration
1022	TRN_FLT_APP	Terrain Flight Approach
1024	HVR_AUTOROT	Hovering Autorotation
1026	SIM_ENG_FAIL	Simulated Engine Failure
1028	MAN_THROTTLE	Manual Throttle Operation
1029	NO_SCAS/SAS	SCAS Off Flight
1039	MSKG/UNMSKG	Masking and Unmasking
1042	MOVE_TECH	Techniques of Movement
1048	PIN/RDG_OPN	Pinnacle/Ridgeline Operation
1054	OP_RMS	Operate Rocket Mgmt. System
1055	WPNS_PROC	Weapons Cockpit Procedure
1056	OP_TURRET	Operate 197 Turret System
1057	OP_FFAR_ROC	Operate 2.75" FFAR Launcher
1058	OP_TOW_MSL	Operate TOW Missile System
1061	NOE_FIRING	Terrain Flight Firing Techs.
2000	STD_AUTOROT	Standard Autorotation
2003	S/A_TRQ_MAL	Simulated Antitorque Malfunc.
2004	L-A,H-S_AR	Lo Level Hi Speed Autorotation
3008	DROOP_CMP_FL	Droop Compensator Failure
3010	TAIL_ROT_FL	Tail Rotor Failure at a Hover
3011	MAST_BUMPING	Mast Bumping
3025	THROTTLE_FL	Emergency Shutdown--Throttle
3030	LNDG/TREES	Landing in Trees
3033	SCAS_FAILURE	SCAS Failure

4. the type of motion systems (if any),
5. the type of extra-cockpit visual systems (if any),
6. the type of aerodynamic models (if any), and
7. the type of noise simulated (if any).

Once the generic design characteristics of the family of training devices had been specified, the SMEs proceeded through

the master task list (described above) and identified the tasks that they judged could be trained effectively in each device.

Analysis Overview

Our concept of operation for the OSBATS Model is based on the iterative use of the five model tools to make recommendations regarding the definition of task clusters, the design of training devices for individual clusters, and the allocation of training resources among training devices. This iterative concept of use reflects the interactions inherent in the training-concept development, training-device design, and resource allocation components of the training-system design process. Both the subset of tools that are used and the order in which they are used may vary depending on the requirements of the problem and the preferences of the user. In addition, a tool may be used several times, each time producing a more refined solution.

For the purposes of this example, however, we apply the tools in the following order:

1. The Simulation Configuration module defines clusters of tasks for actual equipment (AE), a full-mission simulator (FMS), and part-mission simulators (PMSs).

2. The Instructional Feature Module focuses on the FMS tasks defined by the Simulator Configuration Module and identifies the instructional features that produce the greatest improvement in training efficiency given their development cost.

3. The Fidelity Optimization Module identifies the areas in which investment in additional fidelity would be cost efficient. The module incorporates the results of the Instructional Feature Module to produce two candidate FMS designs at different development cost levels.

4. The Training-Device Selection Module is applied to evaluate training systems with each of the two candidate devices generated by the Fidelity Optimization Module. For each of these devices, the Training-Device Selection Module determines the minimum cost to meet the training requirements.

5. The Resource Allocation Module determines the minimum cost to meet training requirements using the best training device determined by the Training-Device Selection Module. The Resource Allocation Module takes into account hypothetical constraints on the number of training devices that may be obtained.

In our description of the OSBATS model, we point out alternative approaches that could be taken to the application of the model, opportunities for iteration of the model, and ways the model could be used to refine the solution that is obtained in the analysis.

Analysis Details

The following sections describe each of the modules in turn, and illustrate the kinds of data that are considered in each analysis, the procedures by which data are combined to produce a recommendation, and the particular recommendations that are made by each module. The procedures are described at a general level. The reader who wishes a more detailed description of the procedures will find them in the formal model description in a later section. The goal of this section is to provide a description of the model from a user's perspective; thus, it concentrates on the types of recommendations produced by the model, how user judgment is incorporated into the model results, and how the results can be used to develop and evaluate training-system designs.

Simulation Configuration Module

The Simulation Configuration Module examines the tasks to be trained and partitions the task set into three clusters: (a) a cluster of tasks that should be trained on a full-mission simulator (FMS) that represents most of the actual equipment subsystems, (b) a cluster of tasks that should be trained on one or more part-mission simulators (PMSs) that represent selected subsystems or that represent actual equipment at a low level of fidelity, and (c) a cluster of tasks that should be trained on actual equipment (AE). This recommendation is based upon simulator requirements for task training and the potential operating-cost savings associated with simulators, balanced against simulator development cost, as estimated by the task cue and response requirements.

Initial model recommendations. In this example problem, we assume that all relevant data have been collected and are present in the OSBATS data base. The Simulation Configuration Module can then make an immediate recommendation when the user has specified the set of tasks to be trained. In the example problem, the model recommends that 22 of the 38 tasks be trained on a full-mission simulator, 11 tasks be trained on part-mission simulators, and 5 tasks be trained on actual equipment. The initial recommendations of the module are shown in Table 8.

These recommendations are based on the following assumptions:

1. Simulation requirements and operating cost savings are weighted equally in determining the indications for simulation, the measure of simulation benefit used in the module.

Table 8 Initial Recommendations of the Simulation Configuration Module

Actual Equipment Task #	Name	Part-Mission Simulator Task #	Name	Full-Mission Simulator Task #	Name
1012	DOPPLER_NAV	1003	T/O_CHECKS	1015	SHLW_APPR
		1004	HOVER_PWR_CK	1016	CONFIND_OPN
		1005	HOVER_FLT	1019	TERRAIN_FLT
		1006	NORMAL_T/O	1042	MOVE_TECH
		1007	S/MAX_P_T/O	1048	PIN/RDG_OPN
		1008	DECEL_ACCEL	1054	OP_RMS
		1009	TRFC_PTRN	1056	OP_TURRET
		1013	PRE_LNDG_CK	1057	OP_FFAR_ROC
		1014	VMC_APPROACH	1058	OP_TOW_MSL
		1018	TERR_FLT_T/O	1061	NOE_FIRING
		1021	NOE_DECEL	2000	STD_AUTOROT
		1022	TRN_FLT_APP	2003	S/A_TRQ_MAL
		1024	HVR_AUTOROT	2004	L-A,H-S_AR
		1026	SIM_ENG_FAIL	3010	TAIL_ROT_FL
		1028	MAN_THROTTLE	3033	SCAS_FAILURE
		1029	NO_SCAS/SAS		
		1039	MSKG/UNMSKG		
		1055	WPNS_PROC		
		3008	DROOP_CMP_FL		
		3011	MAST_BUMPING		
		3025	THROTTLE_FL		
		3030	LNDG/TREES		

2. The total operating-cost savings calculated by the module is used as the basis of the cost boundary that divides AE tasks from PMS and FMS tasks.

3. The task complexity line, which is used to separate FMS tasks from PMS tasks, is located at the halfway mark on the normalized cost scale.

In this hypothetical problem, the user will examine the assumptions of the model to verify the accuracy of the results. The examination begins with general characteristics of the model results and proceeds to more specific features.

<u>Examination of model recommendations</u>. The first aspect of the results to examine is the ordering of tasks based on cost and indications for simulation. Table 9 lists tasks that have extreme values on either cost or indications for simulation. This display is not presented directly by the software, but could be generated easily by the user from other displays. We anticipate that the user will be able to verify the tasks on this list given his general knowledge of the training domain. More detailed information would be available to the user if it were

Table 9 List of Tasks with Extreme Values of Cost or Indications for Simulation

Tasks with High Indication for Simulation	
1061	NOE FIRING
3008	DROOP CMP FL
3010	TAIL ROT FL
3011	MAST BUMPING
3025	THROTTLE FL
3030	LDNG/TREES
Tasks with Low Indication for Simulation	
1012	DOPPLER NAV
1013	PRE LNDG CK
Tasks with High Cost	
1012	DOPPLER NAV
1042	MOVE TECH
3033	SCAS FAILURE
Tasks with Low Cost	
1003	T/O CHECKS
1013	PRE LNDG CK
1055	WPNS PROC
3011	MAST BUMPING
3025	THROTTLE FL

needed to provide a general verification of the cost and indications-for-simulation scales.

The module results shown in Table 9 indicate that six tasks require simulation. An examination of these tasks indicates that this result is reasonable; these tasks involve serious malfunctions (droop compensator failure or tail rotor failure), require use of expensive ammunition or large training areas (terrain firing), or involve dangerous emergency procedures (landing in trees). The two tasks for which indications for simulation has the lowest value also seem reasonable. These tasks involve basic maneuvers that can be learned relatively quickly with little chance of danger.

The cost of simulation reflects the cue and response requirements of the tasks. The tasks with the highest cost require a sophisticated visual system, a sophisticated motion platform or a large data base to adequately train navigation and movement techniques. The tasks with lowest cost are generally procedural, and could be trained on a cockpit procedures trainer with low fidelity.

The preliminary examination of the results indicates that the cost and indication-for-simulation scales reasonably reflect the actual cost and benefits of simulation for the subject tasks. In the next phase of the analysis, the user examines the cost boundary and the task complexity line. The cost boundary separates tasks assigned to the AE cluster from tasks assigned to the FMS and PMS clusters. The task complexity line separates the PMS from the FMS task clusters. That investigation proceeds by examining tasks near these boundaries to determine whether they appear to be in the correct task cluster. Making these close comparisons requires the user to examine the data that underlie the cost and benefit assessments.

Table 10 illustrates the data and intermediate results that underlie the overall module results. The user has access to this information for each task and could verify that for most of the tasks there is essentially no cost savings associated with simulator training, though there may be a moderate value for the indication for simulation. The exceptions to this are tasks 1055, 1057 and 1061, which have a high cost savings. In light of these data, the user might make one of following judgments.

1. The moderate value of the indication for simulation for many of the tasks may lead the user to accept the initial result of the module, that most of the tasks be trained in a simulator.

Table 10 Data Underlying Results for Task # 1057: Operate 2.75" FFAR Launcher

Weights: Training Requirements Index: 0.5
Operating-Cost Savings Index: 0.5

Relative Cost = 0.88

Cue & Response Dimension	Rqmt.	Cost
Visual Resol.	0.98	2638
Visual Content	0.58	496
Visual Texture	0.84	1108
Visual Front	0.86	403
Visual Side	0.46	10
Point Effects	0.00	504
Area Effects	0455	147
Platform Mot	0.00	0
Seat Motion	0.36	67
Sound Effects	0.54	96
Map Size	0.58	634
Total Cost		6106

Indication for Simulation = 0.20

Factor	Value	Wt.
Training Rqmt Index	1.00	0.50
Absolute Requirement	no	
Special Conditions	3 of 3	
Training Effects	2 of 2	
Ops Cost Saving Index	0.64	0.50
Ops Cost Savings	$2505 /student	
AE Hours Saved	1.41	
Device Hours Used	10.28	

2. The user may think that tasks with low cost savings should be trained on actual equipment. The user might then raise the weight given to the operating-cost savings index in the determination of the indication for simulation, to make cost savings a more important consideration in the module.

3. The user may determine that for tasks with low cost savings the training requirements justify simulator training. The user might then raise the weight given to the training requirements index in the determination of the indication for simulation to make the cost savings a less important consideration in the module.

4. By examining other tasks, the user may decide that the model has underestimated the cost savings that would actually occur. In this case, the user might adjust the slope of the cost boundary downward in proportion to the degree to which cost savings were underestimated, thereby including more tasks in the FMS task cluster.

In this example, we assume that the user makes the second judgment listed above. If the user increases the relative weight for the operating-cost savings index from 0.5 to 0.9, the value of the indication for simulation for most tasks decreases. Four new tasks are added to the actual equipment task cluster (Figure 6).

Figure 6 is a representation of the graphical summary of results presented by OSBATS when the weights of the training requirements index and operating-cost savings index have been adjusted. Each asterisk (*) represents a task. If two tasks occupy the same point they are represented by a number two (2). We assume that the user agrees with the tasks assigned to the FMS cluster, but thinks that the task Simulated Engine Failure might be trained better in an FMS rather than in a PMS because it requires numerous systems. We also assume the user thinks that the task Manual Throttle should remain in the PMS task cluster. The task complexity line could be decreased from 0.5 to 0.4 on the normalized cost scale to achieve the desired changes in assignment of tasks to task clusters.

The results of the change in location of the Task Complexity Line are shown in Figure 7. Notice that three tasks, Simulated Engine Failure, VMC Approach (1014) and Terrain Flight Approach (1022), moved from the PMS task cluster to the FMS task cluster. The task Manual Throttle Operation remains in the PMS task cluster.

In addition to changing the location of the task complexity line and changing the relative weights of the training requirements index and the operating-cost savings index, the user can adjust the cost boundary slope. To assist the user in adjusting the cost boundary slope there is a display called Sensitivity Analysis.

Training Requirements Index =0.1   Cost Boundary Slope =Base
Operating Cost Savings Index =0.9   Task Complexity Line =0.5

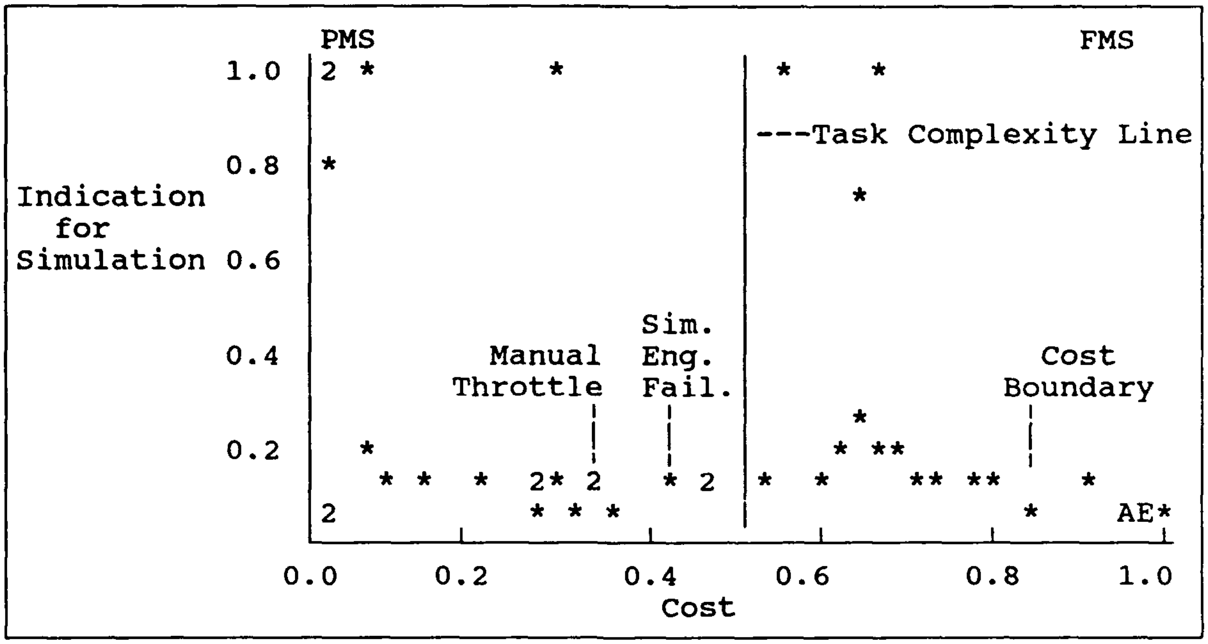


Figure 6. Graphical summary of the Simulation Configuration Module results.

Training Requirements Index ≈0.1   Cost Boundary Slope =Base
Operating Cost Savings Index ≈0.9   Task Complexity Line =0.4

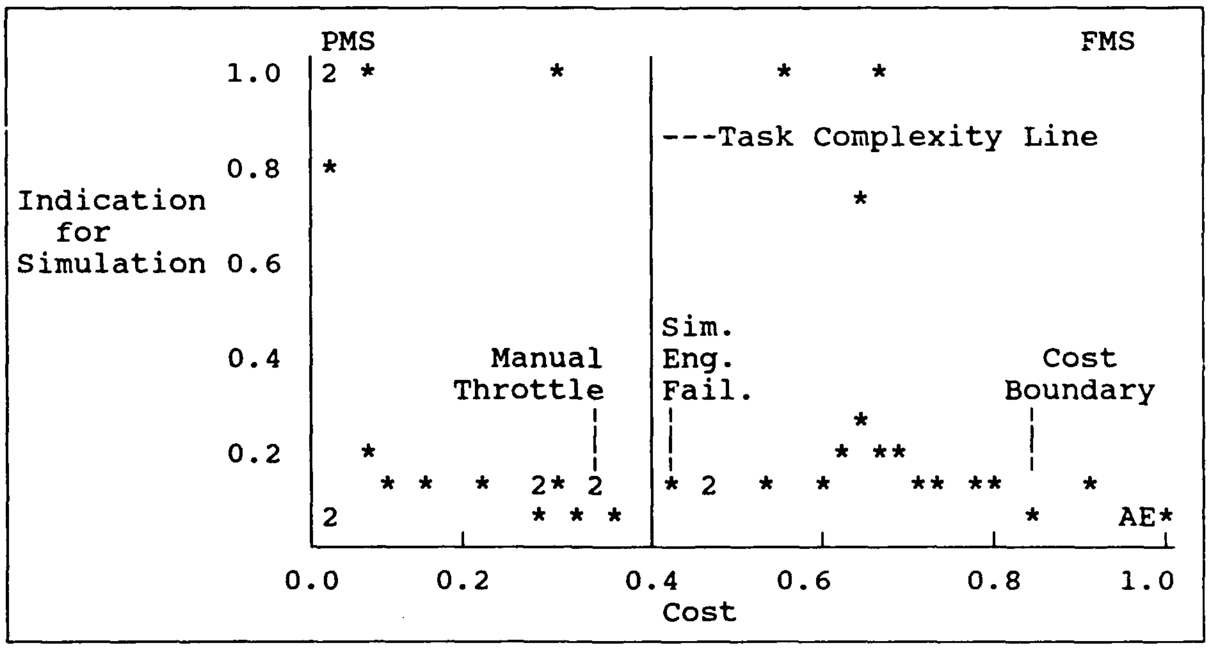


Figure 7. Graphical summary of final recommendations of the Simulation Configuration Module.

Sensitivity analysis and selection of final solution. The final step in the analysis is an examination of the cost boundary and how it affects the task clusters. This is done through a sensitivity analysis, as presented in Table 11, in which the boundary is varied from 40% below to 40% above its nominal value. A low slope, which indicates that the cost savings has a higher weight than the development cost, causes more tasks to be included in the FMS task cluster than a high slope.

When the user has completed the examination of the results of the Simulation Configuration Module, the cluster definitions can be saved to be used by other modules. The final results of the module are shown in Table 12. These clusters form the requirements for the design of training devices, and are used by

Table 11 Number of Tasks in Each Cluster as Cost Boundary Slope is Varied

	Slope of Cost Boundary				
Task Cluster	-40%	-20%	base	+20%	+40%
FMS	17	17	15	13	11
PMS	19	19	18	18	17
AE	2	2	5	7	10

Table 12 Final Recommendations of the Simulation Configuration Module in Tabular Summary

Actual Equipment		Part-Mission Simulator		Full-Mission Simulator	
Task #	Name	Task #	Name	Task #	Name
1004	HOVER_PWR_CK	1003	T/O_CHECKS	1014	VMC_APPROACH
1012	DOPPLER_NAV	1005	HOVER_FLT	1015	SHLW_APPR
1016	CONFIND_OPN	1006	NORMAL_T/O	1019	TERRAIN_FLT
1042	MOVE_TECH	1007	S/MAX_P_T/O	1022	TRN_FLT_APP
3033	SCAS_FAILURE	1008	DECEL_ACCEL	1026	SIM_ENG_FAIL
		1009	TRFC_PTRN	1048	PIN/RDG_OPN
		1013	PRE_LNDG_CK	1054	OP_RMS
		1018	TERR_FLT_T/O	1056	OP_TURRET
		1021	NOE_DECEL	1057	OP_FFAR_ROC
		1024	HVR_AUTOROT	1058	OP_TOW_MSL
		1028	MAN_THROTTLE	1061	NOE_FIRING
		1029	NO_SCAS/SAS	2000	STD_AUTOROT
		1039	MSKG/UNMSKG	2003	S/A_TRQ_MAL
		1055	WPNS_PROC	2004	L-A,H-S_AR
		3008	DROOP_CMP_FL	3010	TAIL_ROT_FL
		3011	MAST_BUMPING		
		3025	THROTTLE_FL		
		3030	LNDG/TREES		

the Instructional Feature Selection Module and the Fidelity Optimization Module. These two modules are described in the following sections.

Instructional-Feature Selection Module

The Instructional-Feature Selection Module begins the training-device design process. The training-device design process is conducted for each training device incorporated into the training system. Any task cluster may form the requirements for a training device. The user may use any of the task clusters produced by the Simulation Configuration Module to start the training-device design process, or the user may apply the Instructional-Feature Selection Module to a task cluster of his or her own making. The Instructional-Feature Selection Module uses a set of rules, which have been developed through research, to compute the benefit on a task-by-task basis for each available feature. The feature benefits are compared to their respective costs and a cost-benefit priority list of features is developed. This priority list can then be incorporated into the Fidelity Optimization Module so that the instructional features can be prioritized with training-device fidelity options.

In this example, we will evaluate the 21 candidate instructional features shown in Table 13 according to how well they can be applied in a training-device to train the FMS task cluster produced by the Simulation Configuration Module. The results of this model will order the candidate instructional features from most to least useful according to the ratio of benefit to development cost. The user will then select several instructional features that will be investigated further in the Fidelity Optimization Module.

Initial model solution. The initial recommendations of the Instructional-Feature Selection Module are shown in Table 14. The results indicate that the first three instructional features, which have the highest benefit-to-cost ratio, offer significant increase in training efficiency considering the cost, the next ten features offer considerably less benefit, and the eight remaining instructional features offer very little or no benefit.

Figure 8 shows graphically the cost and benefit of optimal packages of instructional features. The first non-zero point on the figure represents the cost and benefit of the instructional feature with the highest benefit-to-cost ratio. In this example, the first point represents the feature Reset/Reposition. The second point represents the cumulative cost and benefit of the two features having the highest benefit-to-cost ratios, and so forth. The final point represents the cumulative cost and benefit of a package containing all the instructional features.

Table 13 Instructional Features Evaluated by Instructional-Feature Selection Module

Instructional Feature	Abbreviation
Automated Performance Measurement	perfmeas
Performance Indicators	perf ind
Procedure Monitoring	proc mon
Automated Performance Alerts	perfalrt
Augmented Feedback	aug fdbk
Augmented Cues	aug cues
Record/Playback	rec/play
Total System Freeze	sys frez
Remote Graphics Replay	graph rp
Initial Conditions	init cnd
Scenario Control	scen ctl
Crash Override	crash ov
Reset/Reposition	reset/re
Parameter Freeze	parm frz
Flight System Freeze	flt frez
Positional Freeze	pos frez
Real-Time Simulation Variables Control	realtime
Automated Simulator Demonstration	autodemo
Adjunct CAI	adjn cai
Automated Adaptive Training	adaptive
Automated Cueing and Coaching	coaching

The optimal package of instructional features can be determined from this graph as a function of the budget for instructional features. For example, if the budget for instructional features were $150K then the model would recommend incorporating four instructional features in the training device: Flight System Freeze, Crash Override, Total System Freeze, and Reset/Reposition. Of course, usually there will not be a budget designated for instructional features. In this case, several optimal packages of instructional features can be chosen for evaluation in the Fidelity Optimization Module.

<u>Examination of the solution</u>. The cost of each instructional feature represents development cost and is taken directly from the data base. The benefit, on the other hand, is calculated from the number of tasks for which that instructional feature was estimated to improve training efficiency. The user can examine which instructional features are appropriate for each task to verify the accuracy of the model estimation procedures. The module assigns features to tasks, as shown in Table 15, according

Table 14 Initial Recommendations of the Instructional-Feature Selection Module

Instructional Features	Tasks Wts.	Ben. Wt.	Benefit	Norm. Ben.	Cost (000's)	Norm. Cost	B/C
Reset/Repos.	89	4.63	4.16	0.11	30	0.01	7.39
Total System	89	3.10	2.78	0.07	24	0.01	6.17
Crash Overri	49	4.77	2.38	0.06	29	0.01	4.37
Flight Syste	41	3.80	1.58	0.04	35	0.02	2.40
Initial Cond	99	4.30	4.30	0.11	98	0.05	2.34
Procedure Mo	92	3.40	3.16	0.08	74	0.04	2.27
Positional F	41	3.30	1.37	0.03	35	0.02	2.08
Parameter Fr	41	2.95	1.22	0.03	35	0.02	1.86
Scenario Con	99	4.40	4.40	0.11	155	0.07	1.51
Real-Time Si	99	3.70	3.70	0.09	137	0.07	1.44
Remote Graph	75	2.70	2.03	0.05	76	0.04	1.42
Automated Si	41	2.16	0.90	0.02	54	0.03	0.88
Record/Playb	89	1.42	1.27	0.03	92	0.04	0.74
Automated Pe	99	1.40	1.40	0.04	215	0.10	0.35
Adjunct CAI	89	1.70	1.53	0.04	236	0.11	0.34
Automated Cu	89	1.10	0.99	0.03	156	0.07	0.34
Automated Ad	99	1.59	1.59	0.04	253	0.12	0.33
Augmented Cu	39	1.30	0.59	0.01	155	0.07	0.17
Augmented Fe	9	1.30	0.13	0.00	97	0.05	0.07
Performanc I	0	1.40	0.00	0.00	51	0.02	0.00
Automated Pe	0	1.30	0.00	0.00	61	0.03	0.00

to a set of rules that make assignments depending on basic task information-processing variables. The rules are not accessible to the user in the current version of the software. The assignment of tasks to an instructional feature is illustrated in Table 16.

The overall instructional feature benefit is related to the number of tasks for which the instructional feature is appropriate. However, all tasks do not receive equal weight in determining benefit. Tasks that require more training receive a higher weight in the benefit calculation. The task weights are based on the total cost of training the tasks to criterion. In addition, empirical research has indicated that some instructional features are used by instructors more often than others even though they may be equally appropriate. For this reason the weighted average benefit calculated by the model is moderated by a feature weight that depends only on the instructional feature. Examination of the task weights and

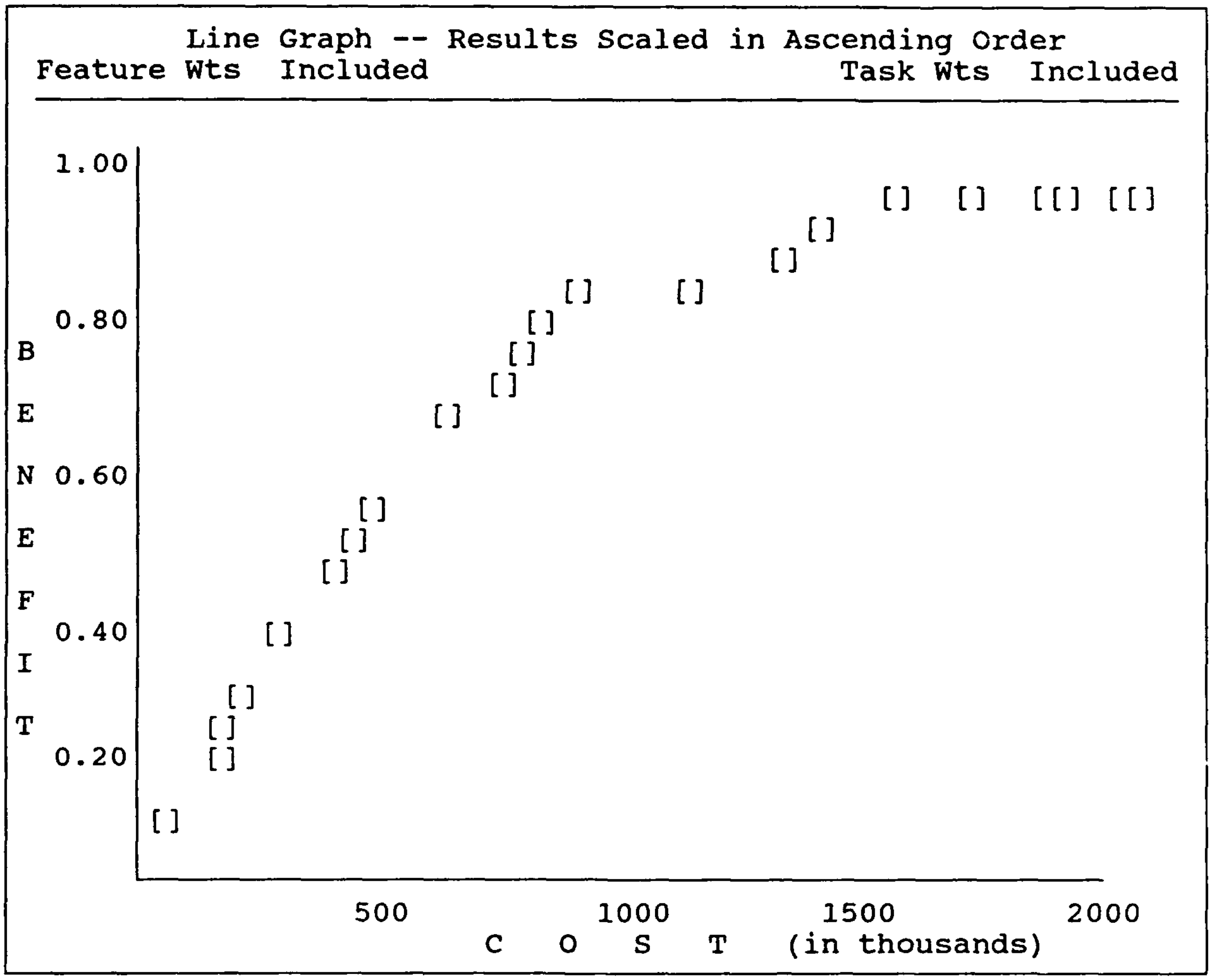


Figure 8. Graph of cost and benefit of optimal instructional-feature packages.

feature weights can give an indication of the rationale behind the benefit values.

There are conditions in which the user may feel that it is inappropriate to use task weights or feature weights. For example, if the goal of a training device is to provide familiarization training on a variety of tasks, the task weights, which are based on training tasks to criterion, may not be appropriate. In this case, an unweighted average of tasks may more accurately represent the benefit of an instructional feature. Similarly, since the feature weights are essentially a measure of instructor bias, they may be counteracted by training, familiarization with the instructional feature, or motivational incentives. Thus, the OSBATS user may feel that these biases should not be considered in determining instructional-feature benefit.

Table 15 Assignment of Instructional Features to Task 1014, VMC Approach

Feature Wts Included	Task Wts Included
IFs used in training Task Number 5	
Automated Performance Measurement	1.4
Initial Conditions	4.3
Scenario Control	4.4
Real-Time Simulation Variables Control	3.7
Automated Adaptive Training	1.6

Table 16 Assignment of Tasks to the Instructional Feature, Automated Performance Measurement

Feature Wts Included — Task Wts Included

Tasks and the Number of IF's Used in Training

ATM	Name	Weight
1014	VMC Approach	1.90
1015	Shallow Approach/Landing	1.90
1019	Terrain Flight	1.01
1022	Terrain Flight Approach	0.51
1026	Simul. Engine Fail. -Altitud	0.63
1048	Pinnacle/Ridgeline Operation	1.80
1054	Operate Rocket Mgmt. System	9.81
1056	Operate 197 Turret System	7.94
1057	Operate 2.75" FFAR Launcher	21.26
1058	Operate TOW Missile System	9.21
1061	Terrain Flight Firing Techs.	36.68
2000	Standard Autorotation	2.41
2003	Simulated Antitorque Malfunc	2.41
2004	Lo Level Hi Speed Autorotati	2.41
3010	Tail Rotor Failure at a Hove	0.12

Table 17 compares the original solution with the solutions obtained when the task weights and the feature weights are excluded. Only the 11 instructional features with the highest benefit-to-cost ratios in each solution are shown. The results are similar in many respects, but there are noticeable differences between the solutions.

When the task weights are eliminated, the feature named Crash Override jumps from third to first in the order. Examination of which features are assigned to each task gives an indication as to why this change occurs. The user could verify

Table 17 Comparison of Solutions when Task Weights and Feature Weights are Ignored

Both Weights Considered	Task Weights Ignored	Feature Weights Ignored
reset/re	crash ov	sys frez
sys frez	reset/re	reset/re
crash ov	sys frez	crash ov
flt frez	init cnd	proc mon
init cnd	proc mon	pos frez
proc mon	scen ctl	parm frz
pos frez	real tim	flt frez
parm frz	flt frez	init cnd
scen ctl	pos frez	graph rp
real tim	parm frz	rec/play
graph rp	graph rp	autodemo

from the Instructional Feature - Task Assignment Lists that there are 10 tasks for which Crash Override is appropriate, while there are 8 for Reset/Reposition and 7 for Total System Freeze. However, the tasks assigned to Crash Override involve a relatively smaller amount of training to reach criterion. Reset/ Reposition and Total System Freeze are appropriate for the difficult weapons tasks that require much more training. Given this comparison, we will assume that the user accepts the original solution, because the FMS would be expected to take a substantial amount of the training load for the weapons tasks.

When the feature weights are set equal to each other, the feature Total System Freeze moves from second on the list to first. An examination of the feature weights found in Table 14 indicates why this result occurs. The feature Reset/Reposition was judged by instructors to be very useful, and consequently has a high benefit weight. Total System Freeze, on the other hand, was judged less likely to be used, and received a weight approximately two-thirds that of Reset/Reposition. In the original calculations, Total System Freeze received a lower overall benefit than it otherwise would have received. When feature weights are excluded, Total System Freeze has a relatively higher benefit-to-cost ratio than Reset/Reposition because it is less expensive.

<u>Selecting and saving instructional-feature alternatives</u>. We assume that the user accepts the solution with both task weights and feature weights included. The final task for the user is to choose a range of instructional range features to be included in the Fidelity Optimization Module. We will assume for this example that the minimum level the user will consider includes the first three instructional features, Reset/Reposition, Total System Freeze and Crash Override ($83K). The user is willing to consider the optional addition of the

following features to the minimum package. As a second level, the user may consider the first three features plus Flight System Freeze, Initial Conditions, Procedure Monitoring and Positional Freeze ($325K). The third level includes the first and second levels plus Parameter Freeze and Scenario Control ($515K). The highest level of instructional features the user will consider is the cumulative package including the first three packages and Real-time Simulation Variables Control, Remote Graphics Replay, Automated Simulation Demonstration and Record/Playback ($874K). These four options (the minimum option plus the three cumulative options) will then be incorporated in the Fidelity Optimization Module to obtain an overall training-device design.

Fidelity Optimization Module

The Fidelity Optimization Module continues the training-device design process begun with the Instructional Feature Selection Module. The goal of the Fidelity Optimization Module is to specify the levels of technical sophistication on each of a set of fidelity dimensions to maximize the effectiveness of a training device given its development cost. The Fidelity Optimization Module may recommend several optimal training-device designs at different costs. Each of the recommended designs could then be evaluated by the Training-Device Selection and Resource Allocation Modules. A second function of the Fidelity Optimization Module is to make tradeoffs between technical sophistication in presenting cues and response feedback and technical sophistication in instructional support. To accomplish this goal the Fidelity Optimization Module considers solutions to the Instructional Feature Selection Module as an additional variable in its evaluation.

In this example, we will evaluate eleven candidate fidelity dimensions to design a training device for the FMS task cluster. Each of the fidelity dimensions may be represented at a number of levels as shown in Table 18. This model uses data concerning the cue and response requirements for each fidelity dimension to determine the benefit of each level of each dimension. Using cost formulas based upon technical performance, cost measures are also calculated for each fidelity option. Then the fidelity options across all fidelity dimensions are ordered according to the benefit-to-cost ratio and optimal training-device designs are defined for given levels of development cost. The user can then select several simulator designs at desired levels of cost or benefit. These training-device designs can then be analyzed by the Training-Device Selection Module.

Initial model solution. The model solution is based on estimates of cost and benefit for the options. The benefit values, shown in Table 19, are based on the number of tasks that could be trained at each level of each dimension. That is, for a given level of a fidelity dimension, the benefit is based on the number of tasks for which the technical performance of that level is at least as great as the corresponding task cue and response requirement. Tasks are weighted by the task weights, as they

Table 18 Dimensions and Levels Considered by the Fidelity Optimization Module

---

Dimension 1 - Visual Resolution

Level 1 - M2 at 0.3 km
Level 2 - M2 at 0.5 km
Level 3 - M2 at 1.0 km
Level 4 - M2 at 2.0 km
Level 5 - M2 at 3.0 km
Level 6 - M2 at 4.0 km

Dimension 2 - Visual Content

Level 1 - Plane with Trees
Level 2 - Add Generic Features
Level 3 - Realistic Density
Level 4 - Low Density Hydrographic features
Level 5 - Medium Density Hydrographic features
Level 6 - High Density Hydrographic features

Dimension 3 - Visual Texture

Level 1 - Lines and Polygons
Level 2 - Modulating Functions
Level 3 - Few Digitized Photographs
Level 4 - More Digitized Photographs
Level 5 - Many Digitized Photographs

Dimension 4 - Front Field of View

Level 1 - 40 x 40 degrees
Level 2 - 40 x 50 degrees
Level 3 - 40 x 60 degrees

Dimension 5 - Side Field of View

Level 1 - 40 x 40 degrees
Level 2 - 40 x 50 degrees
Level 3 - 50 x 50 degrees
Level 4 - 50 x 60 degrees
Level 5 - 40 x 50 degrees - two sides
Level 6 - 40 x 60 degrees - two sides
Level 7 - 50 x 60 degrees - two sides

Table 18 (continued)

Dimension 6 - Special Effects - Points
Level 1 - None
Level 2 - Cultural Lights
Level 3 - Add Weapons Blast
Level 4 - Add Damaged Vehicles
Level 5 - Add Airborn Vehicles
Level 6 - Add Moving Ground Vehicles
Dimension 7 - Special Effects - Area
Level 1 - None
Level 2 - Smoke and Dust
Level 3 - Rotor Wash
Dimension 8 - Motion Platform
Level 1 - None
Level 2 - 3 degrees of freedom
Level 3 - 5 degrees of freedom
Level 4 - 6 degrees of freedom
Dimension 9 - Seat Motion
Level 1 - Stationary
Level 2 - Seat Shaker
Level 3 - Add G-Seat
Dimension 10 - Audio Effects
Level 1 - None
Level 2 - Weapons, Skid, Failures
Level 3 - Add Normal Operating Noise
Level 4 - Add Abnormal Operating Noise
Dimension 11 - Data Base Size
Level 1 - 5 x 5 km
Level 2 - 10 x 10 km
Level 3 - 10 x 20 km
Level 4 - 10 x 30 km
Level 5 - 20 x 30 km
Level 6 - 30 x 30 km
Level 7 - 30 x 40 km

were in the Instructional Feature Selection Module. Each fidelity dimension receives a weight that depends on the overall pattern of cue and response requirements for that dimension. If many tasks require high levels of technical performance on a dimension, then the dimension will receive a high weight in determining overall benefit of a training-device design.

The estimated costs are shown in Table 20. These costs are estimated directly from technical performance using an equation that assumes that low levels of technical performance can be obtained relatively cheaply compared to levels of performance that are closer to the state of the art. The cost estimates represent development cost only.

The incremental benefit-to-cost ratio for the fidelity-dimension levels is shown in Table 21. In calculating the incremental benefit-to-cost ratio, the module eliminates those options that are not cost-efficient within each dimension. For example the benefit-to-cost ratio for the increment from the first to the second level of the dimension Sound Effects is less than the that for the increment from the second level to the third. In this case, the module eliminates the second level from consideration. It then recalculates the benefit-to-cost ratio for the third level to represent the increment from the first level to the third level.

Selection of candidate training-device designs. The plot of the cost and benefit of optimal designs is shown in Figure 9. The user may access these designs by specifying a target cost. The module will then determine the design that has the greatest benefit at that cost or less. For this example problem, we assume that the user has a development budget of $7 million, but that he would like to investigate the cost-effectiveness of a significantly less expensive design that would cost near $3 million.

When the user specifies the target cost of $7 million, the system specifies the design shown in Table 22. This design has an estimated development cost of $5.5 million, which is below the budget. The user might then examine the next most expensive optimal design, shown in Table 23. This design is considerably over the budget, with a cost of $7.8 million. Examination of the two designs shows that the more expensive design includes a significantly better Visual Resolution system than the less expensive design. The cost of the enhancement is the difference between $2391K (Level 5) and $50K (Level 1), or $2.3 million. Given these estimates, the user would probably wish to examine whether there are other useful ways to fill the $1.5 million gap between the recommended solution, a $5.5 million design, and the $7 million budget.

Table 19 Benefit Values of Fidelity Levels

Level	Benefit 1	2	3	4	5	6	7
Visual_Resol.	0.00	0.40	0.40	1.20	2.92	2.92	
Visual_Content	0.00	3.68	3.73	3.73	3.73	3.73	
Visual_Texture	0.36	1.60	1.60	14.38	14.38		
Visual_Front	0.01	9.57	10.49				
Visual_Side	1.86	1.86	2.10	2.13	4.88	4.88	4.97
Point_Effects	1.77	2.37	2.37	2.37	2.37	15.70	
Area_Effects	1.18	7.81	7.81				
Platform_Mot.	0.75	0.77	0.81	0.81			
Seat_Motion	0.00	7.57	8.17				
Sound_Effects	0.00	0.00	10.71	11.62			
Map_Size	0.09	3.63	3.63	3.63	3.66	3.66	3.66
Inst_features	1.69	3.58	4.60	6.03			

Table 20 Cost (in thousands) of Fidelity Levels

Level	Costs 1	2	3	4	5	6	7
Visual_Resol.	50	887	1744	2235	2391	2471	
Visual_Content	80	496	942	1424	1947	2520	
Visual_Texture	75	393	758	1108	1344		
Visual_Front	20	403	1103				
Visual_Side	10	43	85	121	194	248	319
Point_Effects	0	38	79	167	318	504	
Area_Effects	0	147	336				
Platform_Mot.	0	1173	2084	3024			
Seat_Motion	0	67	192				
Sound_Effects	0	43	96	192			
Map_Size	100	634	918	1216	1855	2564	3360
Inst_features	83	325	515	874			

Table 21 Incremental Benefit-to-Cost Ratios for Fidelity Levels

	Benefit/Cost Ratio						
Level	1	2	3	4	5	6	7
Visual_Resol.	undef	----	----	----	0.12	----	
Visual_Content	undef	0.88	0.01	----	----	----	
Visual_Texture	undef	----	----	1.36	----		
Visual_Front	undef	2.49	0.13				
Visual_Side	undef	----	----	----	1.64	----	0.07
Point_Effects	undef	----	----	----	----	2.76	
Area_Effects	undef	4.51	----				
Platform_Mot.	undef	----	0.00	----			
Seat_Motion	undef	11.16	0.48				
Sound_Effects	undef	----	11.13	0.95			
Map_Size	undef	0.66	----	----	0.00	----	----
Inst_features	undef	0.78	0.54	0.40			

Table 22 Optimal Training-Device Design at a Cost of $5.5 Million

Dimension	Level	Description	Benefit	Cost
Visual_Resol.	1	m2 at 0.3 km	0.0	50.00
Visual_Content	2	Add Generic Featrs	3.7	496.08
Visual_Texture	4	More Digit Ph	14.4	1108.32
Visual_Front	3	40 x 60 Deg	10.5	1103.93
Visual_Side	5	40 x 50 Deg	4.9	194.67
Point_Effects	6	Add Mvng Grnd Veh	15.7	504.00
Area_Effects	2	smoke and dust	7.8	147.22
Platform_Mot.	1	none	0.8	0.00
Seat_Motion	3	Add G-Seat	8.2	192.00
Sound_Effects	4	Add abnor opt nse	11.6	192.00
Map_Size	2	10 x 10 km	3.6	634.40
inst_features	4	Inst Feat, Lev 4	6.0	874.00
Totals			87.1	5496.62

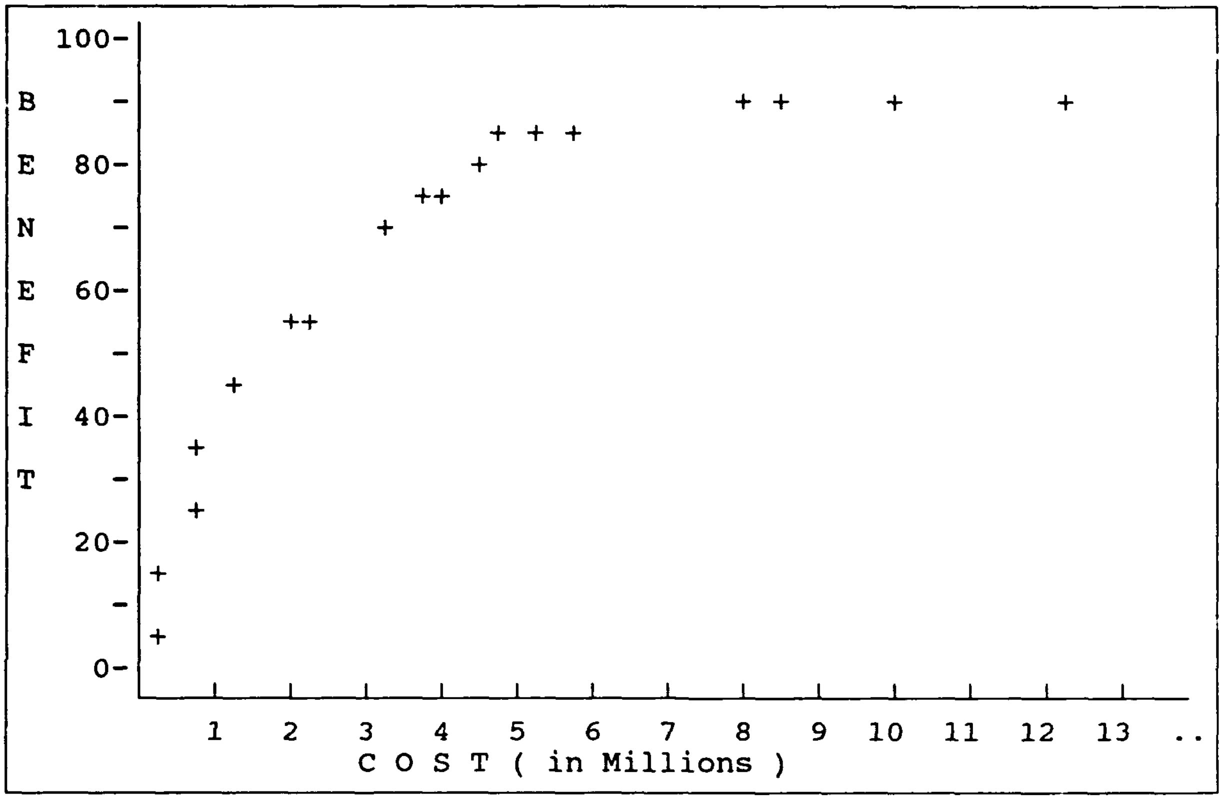


Figure 9. Graph of cost and benefit of optimal training-device designs: Initial analysis.

Table 23 Optimal Training-Device Design at a Cost of $7.8 Million

Dimension	Level	Description	Benefit	Cost
Visual_Resol.	5	m2 at 3.0 km	2.9	2391.12
Visual_Content	2	Add Genric Featrs	3.7	496.08
Visual_Texture	4	More Digit Ph	14.4	1108.32
Visual_Front	3	40 x 60 Deg	10.5	1103.93
Visual_Side	5	40 x 50 Deg	4.9	194.67
Point_Effects	6	Add Mvng Grnd Veh	15.7	504.00
Area_Effects	2	smoke and dust	7.8	147.22
Platform_Mot.	1	none	0.8	0.00
Seat_Motion	3	Add G-Seat	8.2	192.00
Sound_Effects	4	Add abnor opt nse	11.6	192.00
Map_Size	2	10 x 10 km	3.6	634.40
inst_features	4	Inst Feat, Lev 4	6.0	874.00
Totals			90.1	7837.74

This problem may be addressed by eliminating the enhanced visual resolution system from consideration and recalculating the optimal designs. The training-device designs will be recalculated and the $1.5 million formerly designated for visual resolution can be allocated to other options. The results show that this gap is reduced. The optimal design shown in Table 24 is found with a cost of $6.9 million, which is closer to the budget. Note that the recommended design now includes the second level of visual resolution, which costs an additional $837,000 over the first level. It also includes an additional level of the dimension Visual Content, at the additional cost of $450,000, and two additional levels of the dimension Side Field-of-View, for $125,000. The total additional cost over the $5.5 million dollar design is $1.4 million.

We assume the user is satisfied with this design and wishes to save it for further evaluation in the Training-Device Selection module. In addition, the design shown in Table 25 at a cost of $2.9 million is selected. These two designs are then saved to be used as the basis of the analyses performed by the Training-Device Selection Module.

Table 24 Optimal Training-Device Design at a Cost of $6.9 Million

Dimension	Level	Description	Benefit	Cost
Visual_Resol.	2	m2 at 0.5 km	0.4	887.71
Visual_Content	3	Realistic Densty	3.7	942.64
Visual_Texture	4	More Digit Ph	14.4	1108.32
Visual_Front	3	40 x 60 Deg	10.5	1103.93
Visual_Side	7	50 x 60 Deg	5.0	319.68
Point_Effects	6	Add Mvng Grnd Veh	15.7	504.00
Area_Effects	2	smoke and dust	7.8	147.22
Platform_Mot.	1	none	0.8	0.00
Seat_Motion	3	Add G-Seat	8.2	192.00
Sound_Effects	4	Add abnor opt nse	11.6	192.00
Map_Size	2	10 x 10 km	3.6	634.40
Inst_features	4	Inst Feat, Lev 4	6.0	874.00
Totals			87.7	6905.89

Table 25 Optimal Training-Device Design at a Cost of $2.9 Million

Dimension	Level	Description	Benefit	Cost
Visual_Resol.	1	m2 at 0.3 km	0.0	50.00
Visual_Content	1	Plane w/ Trees	0.0	80.00
Visual_Texture	4	More Digit Ph	14.4	1108.32
Visual_Front	2	40 x 50 Deg	9.6	403.27
Visual_Side	5	40 x 50 Deg	4.9	194.67
Point_Effects	6	Add Mvng Grnd Veh	15.7	504.00
Area_Effects	2	smoke and dust	7.8	147.22
Platform_Mot.	1	none	0.8	0.00
Seat_Motion	2	Seat Shaker	7.6	67.88
Sound_Effects	4	Add abnor opt nse	11.6	192.00
Map_Size	1	5 x 5 km	0.1	100.00
Inst_features	1	Inst Feat, Lev 1	1.7	83.00
Totals			74.1	2930.36

<u>Training-Device Selection Module</u>

The Training-Device Selection Module takes as input a set of candidate training devices for the set of tasks. These training devices include full-mission simulators, part-mission simulators, and other training devices designed by the Instructional Feature Selection and Fidelity Optimization Modules. Also included are definitions of existing training devices or templates describing training-device prototypes. Unless the user is considering the purchase of two different full-mission simulators, this module should be exercised separately for each FMS that the user defined in the Fidelity Optimization Module. The Training-Device Selection Module then minimizes the cost to train each task with the devices available, using learning curve data for each task and training-device combination. The output of this model is the aggregated usage of each training device across all tasks. This module does not consider complex cost functions or task interactions in determining the optimal assignment of training devices to tasks; this more complex task is performed by the Resource Allocation Module.

In this example, we will compare the two FMS candidates developed in the Fidelity Optimization Module in a training system that consists of actual equipment, classroom, and other training devices. The complete list of available training devices and their abbreviated names is shown in Table 26. For simplicity, the example analyses will focus on small subsets of these training devices. For each subset of training devices, the module calculates the minimum cost to meet the training requirements using the specified training devices. The user may

then examine which tasks are assigned to which device and the extent to which each device is used to modify the subset of training devices being considered.

The model allocates tasks to training devices by choosing at each level of performance the training device that produces the greatest improvement in performance for the cost. Training continues on this device until a performance level is reached at which the training device is no longer optimal. Training is then switched to the device that is optimal at that performance level. This process continues until the training standard is reached. When training devices have been assigned to all tasks, a summary of training-device use is calculated and presented to the user.

To provide a baseline measure, we will first apply the Training-Device Selection Module to a training system consisting of classroom training and actual equipment only. The model estimates the per-student cost of this training system as $51,900 and the per-student training time as 65.8 hours. The cost value will be used to compare the savings that would occur with the introduction of additional training devices into the system.

Table 26 List of Training Devices Evaluated by the Training-Device Selection Module

Abbreviation	Training-Device Name
CLASSROOM	Classroom Training
CPT	Cockpit Procedures Trainer
DOPPLER	Doppler Navigation Trainer
IMC FT	Instrument Conditions Flight Trainer
VMC FT	Visual Conditions Flight Trainer
WEAPON	Weapon System Procedures Trainer
NOE NAV	Nap-of-the-Earth Navigation Trainer
ACT EQUIP	Actual Equipment
sevenmillion	$7 million FMS from Fidelity Optimization
threemillion	$3 million FMS from Fidelity Optimization

<u>Analysis of the $7 million FMS</u>. We now introduce the $7 million FMS into the system and examine the solution presented in Table 27. These results show an overall per-student cost of $40,500, more than a ten-thousand dollar improvement over the baseline. The $7 million FMS takes over a large amount of the training that was conducted on the actual equipment in the baseline calculations. Though the total hours of training per student (92.8 hours) have increased compared to the baseline, the total cost per student has decreased.

<u>Analysis of the $3 million FMS</u>. In the second part of this analysis the $3 million FMS is included in the training system in place of the $7 million FMS. The results, shown in Table 28,

indicate an overall per-student cost of $37,900, a savings of $2,600 per student compared to the first training system. Though this second training system involves more training hours per student (128 hrs.), the overall cost is lower than the svstem which includes the $7 million FMS. Therefore, the $3 million FMS is superior to the $7 million FMS. The user has access to information about the training devices selected for each task.

The $3 million FMS is used to train more tasks than it was designed for. Of the 15 tasks in the FMS task cluster 12 are assigned training on the FMS. The other three are trained entirely on the actual equipment. In addition, 13 of the 18 tasks in the PMS task cluster are trained partially or fully on the FMS. This result makes sense since we did not design a training device for the task in the PMS cluster. However, the FMS might have greater fidelity than is actually required since its design considered the requirements of three tasks for which it was not used. Consequently, the results of the Training-Device Selection Module have implications for earlier modules.

Table 27 Training-Device Selection Module Results for the $7 Million FMS

Training Devices	Tasks Trained of 38	Cost to Train (000's)	Training Hours / Student	Dev. Rqd.	Hours / Year / Device	Assumed Hours / Dev/Yr
Classroom Training	4	0.19	5.97	3	1593	2000
Actual Equipment	20	19.46	22.49	8	2249	2500
sevenmillion	31	20.87	64.38	18	2860	3000
Totals		40.52	92.84			

Table 28 Training-Device Selection Module Results for the $3 Million FMS

Training Devices	Tasks Trained of 38	Cost to Train (000s)	Training Hours / Student	Dev. Rqd.	Hours / Year / Device	Assumed Hours / Dev/Yr
Classroom Training	4	0.19	5.97	3	1593	2000
Actual Equipment	29	24.82	28.69	10	2295	2500
threemillion	27	12.84	93.31	25	2986	3000
Totals		37.85	127.98			

To further investigate this aspect of the results, the user would define a smaller FMS task cluster to use as the basis for the Instructional Feature Selection and Fidelity Optimization Modules. For example, the user might select to include in the FMS task cluster only those 12 tasks that were assigned training on the FMS. This would result in a smaller FMS task cluster. The training-device designs produced by the Instructional Features and Fidelity Optimization modules would be specific for a smaller number of tasks and therefore would probably have lower fidelity requirements. These designs would then be evaluated by the Training-Device Selection Module to determine if further savings could be obtained.

The user may wish to compare the results obtained using the $3 million FMS with the results obtained from using one of the prototype training devices. There are a number of weapons tasks for which the students have very low entry levels of knowledge. It is possible that the training system would be less expensive if a weapons trainer were used. Since a template for a weapons system trainer exists in the data base, we will add it to the list of devices to replace the current FMS. The results shown in Table 29 indicate that the total per-student cost for this training system is $49,000, more than ten-thousand dollars over the training system with the $3 million FMS.

Table 29 Training-Device Selection Module Results for the Weapon System Trainer

Training Devices	Tasks Trained of 38	Cost to Train (000s)	Training Hours / Student	Dev. Rqd.	Hours / Year / Device	Assumed Hours / Dev/Yr
Classroom Training	4	0.19	5.97	3	1593	2000
Weapons System Trainer	8	2.79	32.39	13	1993	2000
Actual Equipment	33	46.31	53.54	18	2379	2500
Totals		49.29	91.90			

Re-iteration of the Results. The calculations for the per-student cost and per-student time to train the tasks depend on an estimate of the annual usage of each device. Sometimes the value of the calculated annual usage is very different from the estimated value.

The Re-iteration function is used when there is a large discrepancy between the "Assumed Hours/Device/Year" and the actual "Hours/Device/Year." The value of the actual annual usage

is substituted for the estimated annual usage and the module recalculates the results. If the discrepancy is small it is not necessary to use Re-iteration.

Recall that the value of the actual hours used / year / device for the $3 million FMS (2986 hours) was very close to the value that was used to estimate the hourly cost (3000 hours). In this case Re-iteration is not necessary.

<u>Conclusions of the Analysis</u>. It is clear from the analysis that the $3 million FMS is superior to the $7 million FMS. The cost of the $7 million FMS was too high for it to replace significant amounts of actual equipment training at a cost comparable to the cost of the $3 million FMS. For the purposes of this example we assume that the training system that will be analyzed in further detail in the Resource Allocation Module includes the classroom trainer, the $3 million FMS, and the actual equipment.

The results of the Training-Device Selection Module have several implications for reanalysis of the problem by earlier modules. The implications are enumerated below.

1. The Instructional Feature Selection and Fidelity Optimization Modules might be run using a reduced task set. The set might include tasks in the FMS task cluster for which training on the FMS was selected by the Training- Device Selection Module. You may also wish to include those tasks from the PMS task cluster for which a substantial amount of training was selected to be on the FMS by the Training-Device Selection Module.

2. The user might want to design a device based on a task cluster consisting of only the three tasks from the FMS task cluster for which training on the actual equipment was selected by the Training-Device Selection Module. The three tasks are Terrain Flight, Pinnacle Ridgeline Operation and Standard Autorotation. The user might decide to rerun Training-Device Selection and include this design in the current training system to see if the total cost to train each student decreases even further.

3. The current analysis of the Fidelity Optimization Module could be examined to identify other lower-fidelity options that might be more cost-efficient. Since the $3 million solution performed better than the $7 million solution, there may be other candidate designs less expensive than the $3 million simulator that would perform even better.

<u>Resource Allocation Module</u>

The Resource Allocation Module provides results that are similar to those provided by the Training-Device Selection Module. However, these methods differ from those of the Training-Device Selection Module in two important respects. (a)

The Training-Device Selection Module assumes that cost is a simple linear function of training-device use, while the Resource Allocation Module breaks the cost curve into a number of linear segments. (b) The Resource Allocation Module allows the user to specify constraints on training-device use. These constraints specify the maximum time that a training device may be used or the minimum performance level for which a training device may be employed on a specific task.

As a result of its increased accuracy and flexibility, the Resource Allocation Module involves greater computational complexity and calculation time than the Training-Device Selection Module. For the current example problem, the calculations of the Resource Allocation Module can take from two to twenty minutes, depending on the complexity of the problem. This increased computational load limits the extent to which the Resource Allocation Module may be used interactively, although limited interactions are possible.

In the example problem, we apply the Resource Allocation Module twice to the results of the Training-Device Selection Module. The first time, we will not apply any external constraints to the solution. The second time will add constraints and determine their effect on the solution.

Unconstrained solution. Table 30 shows a solution to the same problem that was addressed in the Training-Device Selection Module (see Table 28). Differences in the solution come from the increased accuracy of the cost function. Thus, the Resource Allocation will tend to allocate time to a training device until the device is fully utilized. Then it will not allocate additional time unless there is sufficient time to justify the purchase of an additional device.

Constrained solution. In this analysis, we constrain the hours in which the $3 million FMS may be used. The initial results suggest that 27 such training devices would be required. If we assume that only 12 $3 million FMSs are available, then each student can be allocated a maximum of 45.00 hours on the FMS.

The results of this analysis, shown in Table 31, indicate that the training requirements can be met under the new constraints, although at a somewhat greater cost. The increased per-student cost is the direct result of the addition of the constraint. The mandated reduction in the $3 million FMS was compensated for by increased use of the actual equipment, which is more expensive than the $3 million FMS.

Summary of Example Problem

The sample problem has illustrated how the OSBATS modules interact to make recommendations about the kinds of training devices that should be developed, the specific device designs

Table 30 Unconstrained Solution to Resource Allocation Problem

Training Devices	Tasks Trained of 38	Cost to Train (000's)	Training Hours Student	Dev. Rqd.	Hours / Year
Classroom Training	4	0.20	5.94	3	4752
Actual Equipment	29	24.13	27.80	9	22241
threemillion	27	13.87	99.84	27	79870
Totals		38.20	133.58		

Table 31 Constrained Solution to the Resource Allocation Problem

Training Devices	Tasks Trained of 38	Cost to Train (000's)	Training Hours Student	Dev. Rqd.	Hours / Year
Classroom Training	4	0.20	5.94	3	4752
Actual Equipment	33	34.13	39.00	13	31202
threemillion	20	6.19	44.88	12	35903
Totals		40.51	89.82		

that should be employed, and the extent to which training devices should be used. The five modules performed the following functions.

1. The Simulation Configuration Module organized the training requirements to specify the relative need for training in a simulated environment. In addition, the module recommended which tasks would best be trained by a full-mission simulator rather than a simpler training device.

2. The Instructional Feature Selection Module evaluated a range of instructional support features for a full-mission simulator, and specified the optimal set of features to include at several budget levels.

3. The Fidelity Optimization Module evaluated several critical areas in which the technical capability of a training device to present cues and response feedback could vary, and

recommended the optimal levels of these areas as a function of the training-device investment budget.

4. The Training-Device Selection Module evaluated the training-device designs produced by the Fidelity Optimization Module in the context of the entire set of training requirements. It evaluated the FMS alternatives and specified which tasks would use each available training device to meet training requirements at the minimum cost.

5. The Resource Allocation Module refined the solution of the Training-Device Selection Module to account for complexities in the cost function and constraints in training device availability and use.

This analysis is illustrative and is not complete. The description notes several places where further analysis would be required to arrive at a definitive recommendation. However, the example points out how the OSBATS model can be applied to concept-formulation problems of reasonable complexity.

APPLICATION OF OSBATS TO THE M1 ABRAMS TANK TURRET MECHANICS COURSE

The OSBATS model was designed to be a general tool to aid the training device concept formulation process. However, its only application has been to a sample problem from the domain of aviation. Application to other areas may require changes to data values, model parameters, variables, fidelity dimensions and levels, instructional features, or even to the overall model structure. This section investigates the nature and extent of changes required to apply the OSBATS model to a different domain, specifically that of armor turret maintenance.

The specific domain that will be analyzed in this section is the M1 Abrams Tank Turret Mechanic course (45E10). The proponent for this course is the U.S. Army Ordnance Center and School at Aberdeen Proving Ground, Maryland. However, the course is taught at the U.S. Army Armor School at Ft. Knox, Kentucky. This course differs in many important respects from the AH-1 Airman Qualification Course (AQC) that formed the basis of the initial application of the OSBATS model.

1. The turret mechanic course involves armor, while the AH-1 AQC involves aviation.

2. The turret mechanic course involves maintenance, while the AH-1 AQC involves operation.

3. The turret mechanic course involves initial acquisition of skills, while the AH-1 AQC involves transition of skills to a new weapon system.

4. The turret mechanic tasks are heavily loaded on procedural and cognitive activities, while the AH-1 tasks are heavily loaded on psychomotor activities.

5. The turret mechanic course is taught to enlisted personnel immediately following basic training, while the AH-1 AQC is taught to officers and warrant officers.

The great differences between these two courses provides a good test for the generality of the OSBATS model. In our analysis of the turret mechanic course, we will concentrate on two questions. The first question is whether the OSBATS process will work in a training domain that is considerably different from its original application. The second question is what changes will be required to the data and procedures for successful application of the OSBATS model in the new situation. Our intuitions regarding these questions are that the model will stand unchanged, but will have domain-specific data requirements.

One focus of our comparison of the two training domains will be on whether we can develop general procedures to specify the appropriate data variables, such as fidelity dimensions and

instructional features; determine measurement scales; and assess data values for a new training domain. The extent to which we can develop these procedures will have a great impact on the operating procedures used by the OSBATS model and will to a great extent determine its general applicability.

In the remainder of this section, we will describe the M1 tank turret mechanic course, the data required to analyze this course using the OSBATS model, and the adequacy of the decision process specified by the OSBATS model for providing training device design guidance. First, we will give a general description of the training domain, including the tasks to be trained and the existing training devices. Then, we will describe the requirements for both resident and task data. Finally, we will examine the model process for the turret mechanic course, and summarize the differences between the two applications of the model.

General Description of Training Domain

The training domain selected for this second application of OSBATS is defined by those tasks currently listed in the Program of Instruction (POI) for the M1 Abrams Tank Turret Mechanic Course (643-45E10). The purpose of this entry-level course is to "...train enlisted personnel to perform unit maintenance on the vehicle-mounted armament, associated fire-control, and related systems on the M1/M1A1 tank turret system" (POI 643-45E10, p. 3). The course provides training on 61 of the 122 Skill Level One tasks described in the Soldier's Manual for MOS 45E (STP 9-45E12-SM-TG). The 61 tasks in the present POI comprise both duty position and related technical tasks. As described in the Soldiers Manual, duty position tasks are those that the student must learn thoroughly, because he is expected to perform them on a daily basis. One or more related technical tasks are listed with most duty position tasks. It is assumed that if the student can perform the duty position task, he should also be able to perform any related technical task. Because of this distinction, the Soldier's Manual provides more documentation for duty position as opposed to related technical tasks.

Tasks

The present section describes turret mechanic tasks in terms of two general classification schemes. The section concludes with a description of how these classification schemes were used to select a subset of tasks for further study.

Functional systems. Both the M1 Organizational Maintenance Manual (TM 9-2350-255-20-2-3-1/2/3) and the Organizational Troubleshooting Manual (TM 9-2350-255-20-2-2-1/2/3) classify maintenance tasks according to the three major functional systems of the M1 turret: (a) the hydraulic system, (b) the fire control system, and (c) the electrical system. The POI provides a similar categorization. Each of the 61 POI tasks selected for institutional training is assigned to a block of instruction.

The blocks of instruction are, in turn, clustered into eight annexes or general topics for instruction. The three annexes that provide most of the course content correspond to these three functional systems. Thus, this functional classification of tasks appears to be a well accepted and useful taxonomy.

Skill requirements. Alternatively, turret maintenance tasks can be classified according to their human skill requirements. Examination of task titles revealed four homogeneous categories of tasks. The following categories are similar to some of those described by Aagard and Braby (1976).

1. Perform Procedures. Procedural maintenance tasks are those whose titles use action verbs such as "remove," "install," "replace," and "bleed." These tasks require the mechanic to perform a set sequence of discrete actions with to accomplish some tangible result. Some of the actions require some gross motor skills, but the emphasis in this category is on remembering and performing the actions in the correct sequence.

2. Detect/Classify. This category comprises "check," "adjust," "service," "inspect," and "maintain" tasks. These tasks are very similar to the previous category in that they require the mechanic to perform discrete actions in a prescribed sequence. The unique aspect of these tasks is their intangible purpose, i.e., to detect and classify mechanical faults.

3. Troubleshoot. Troubleshooting tasks require the mechanic to detect, identify, and correct faults in any one of the three major subsystems. The identification of these faults requires the subject to perform some of the detect/classify tasks as described in the previous category, while the prescribed correction may entail certain procedures as described in the first category. The distinction between this category and the previous two is that the task steps are not performed in a set sequence; rather, the task sequence follows complex flow diagrams as presented in the troubleshooting manual. Thus, troubleshooting requires both rule using and decision making skills, which are based on the student's general knowledge of electronics and hydraulics and specific verbal information related to repairing the turret components of the M1. In addition, using the STE-M1 equipment and schematic diagrams requires that the student be able to recognize and identify symbolic information.

4. Prepare forms. Tasks in this category are unlike those in previous categories in that they do not have either a procedural or gross motor skill component. On the other hand, forms are prepared according to specific rules based on specific verbal information either presented on the form or in the block of instruction on maintenance forms. Also, preparation of forms requires the user to be able to recognize and identify symbolic information. However, tasks in this

category are not relevant to the present purposes, because they do not require a device for training.

<u>Task selection</u>. Both classification schemes were used to select six 45E10 tasks for further study. The sample of tasks includes two tasks from each of the three main functional areas in turret maintenance: hydraulic system, fire control system, and electrical system. The sample also includes two tasks from each of the first three skill categories described in the previous section: procedures, detection/classification, and troubleshooting. An additional stipulation was to select only duty position tasks to ensure adequate task documentation. The selected tasks are summarized in Table 32.

In comparison with the other tasks, the troubleshooting tasks are quite extensive in scope. Both the turret electrical and fire control functional systems are divided into multiple subsystems, and each subsystem may evidence multiple fault symptoms. Therefore, it became necessary to limit both tasks to a single symptom in a single subsystem. In addition, there are two types of troubleshooting procedures to consider for most symptoms. Primary troubleshooting procedures are accomplished with the STE-M1 electronic test equipment, which automatically identifies faults and cues an appropriate course of action. Alternate troubleshooting procedures employ the more primitive breakout box to identify specific problems. One primary and one alternative procedure was chosen for each of the two

Table 32 Tasks Selected for Further Study

Title	Number	Skill Type	Functional System
Bleed Air from Elevation Mechanism	091-489-1142	Perform Procedures	Hydraulic
Remove/Install Turret Networks Box	091-489-1010	Perform Procedures	Electrical
Check and Adjust Drift	091-489-1194	Detect/ Classify	Hydraulic
Service Commander's Extension (GPSE)	091-489-1198	Detect/ Classify	Fire Control
Troubleshoot Turret Electrical System	091-489-1056	Troubleshoot	Electrical
Troubleshoot Fire Control System	091-489-1256	Troubleshoot	Fire Control

troubleshooting tasks. Selected symptoms and procedures are described in more detail below.

1. Troubleshoot the Turret Electrical System. The turret electrical system is divided into five subsystems: vehicle/ turret power control, fire control, fan assembly, turret breaker monitor, and panel lights and domelights. The STE-M1 applies to only the first two of these five subsystems. The vehicle/turret power control (V/TPC) subsystem was chosen over the firing circuits subsystem to conceptually distinguish this task from the next one entitled "Troubleshoot Fire Control System." There are five sets of symptoms that are associated with the V/TPC subsystem, each of which may be addressed by either a primary or an alternate troubleshooting procedure. To illustrate a primary troubleshooting procedure, the first symptom entitled "Vehicle Master Power Cannot Be Turned On From Commander's Control Panel" (V/TPC-1) was chosen. For the alternate troubleshooting procedure, the second symptom entitled "Vehicle Master Power Cannot Be Turned Off from Commander's Control Panel" (V/TPC-2) was chosen.

2. Troubleshoot the Fire Control System. The fire control system is divided into six subsystems: auto self test, computer and azimuth/elevation, gunner's primary sight defroster, gunner's auxiliary sight reticle, laser range finder, and thermal imaging system. The auto self test subsystem (ASTS) was chosen because it was the only one to which both primary and alternate troubleshooting procedures applied. To illustrate a primary troubleshooting procedure, the symptom entitled "FIRE CONTROL MALF Light and F Symbol Come On. Computer Manual Self Test Shows No Failure" (ASTS-1) was chosen. For the alternate troubleshooting procedure, the symptom entitled "FIRE CONTROL MALF Light Does Not Come On When A Harness Is Disconnected Or When PANEL LIGHTS TEST Pushbutton Is Pressed" (ASTS-2) was chosen.

Current Training Devices

In addition to classroom instruction, instructors use an actual M1 tank and two different computer-based training devices to provide training on turret mechanic tasks. The tank is used primarily to train tasks related to the turret hydraulic system, whereas the computer-based devices train troubleshooting of both the fire control and electrical systems. The computer-based devices are described in more detail below. This section concludes with a brief summary of the extent to which the devices simulate each of the selected tasks.

Turret trainer. The M1 Tank Turret Organizational Maintenance Trainer (Device 17-67) or, more simply, the turret trainer is a full-scale, three-dimensional representation of an M1 turret. The major physical components of the turret trainer are (a) the turret stand, which is a welded frame structure that supports the M1 turret; (b) the viewing platform, which is a platform mounted at the level of the turret ring for holding up to 13 viewers; and

(c) the instructor's control station, from which the instructor can control the simulation (TM 9-6910-242-10). Each of the turret trainer's 75 exercises simulates a single set of fault symptoms for the student to troubleshoot. The exercises call for the student to identify the symptoms and to use the technical manuals to correct the fault. Many of the procedural tasks (e.g., remove and replace components) can be performed on this trainer exactly as they would be on the actual tank.

Panel trainer. The M1 Turret Organizational Maintenance Troubleshooting Trainer (Device 17-68/1) or, more simply, the panel trainer provides a two-dimensional representation of turret components. The major components include (a) the display panel, which is an upright plane surface containing a combination of pictures, controls, and displays that simulate M1 tank system components, controls, and displays; (b) the viewer assembly, which is a random-access slide projector that presents pictures, word messages, or a combination of both in accordance with actions taken on the display panel and control console; and (c) and instructor/operator station, which includes a CRT terminal and control console for both input and output (TM 9-6910-243-10). Like the turret trainer, each of the panel trainer's 63 exercises simulates a single set of symptoms for the student to troubleshoot. Unlike the turret trainer, the panel trainer's components are represented by two-dimensional pictures whose functions are simulated by rotary and pushbutton switches. Whereas the representations of the STE-M1 and the breakout box are not fully functional measuring devices, they do provide appropriate readings within an exercise.

Simulation of selected tasks. Some of the tasks selected for further study may be trained on the two computer-based devices, while others are either partially supported by the devices or are not supported at all. Table 33 provides a qualitative summary and comparison of the device's capabilities in this regard.

Resident Data Requirements

Resident data refer to those data that are relatively independent of the specific OSBATS application. This statement does not mean that resident data variables will be the same over all training domains. We would not expect to find the same concerns over the environmental visual display in Armor turret maintenance that was so critical in Aviation flight training. Nevertheless, we would expect most of the resident data variables that are relevant for training the pilot of the AH-1 helicopter to be relevant for other helicopters, as well. Differences among

Table 33 Summary of Capability of Devices to Support Training on Selected Tasks

Task Procedure: Symptom	Turret Trainer	Panel Trainer
Bleed Air from Elevation Mechanism	Cannot be performed.	Cannot be performed.
Remove/Install Turret Networks Box (TNB)	Performed as in actual tank.	Simulated by button pushes.
Check and Adjust Drift	Cannot be performed.	Cannot be performed.
Service Commander's Extension (GPSE)	Performed as in actual tank.	Cannot be performed.
Troubleshoot Turret Electrical System		
Primary: V/TPC-1	Simulated by Exercise Numbers 33 and 56.	Simulated by Exercise Number 1.
Alternate: V/TPC-2	Cannot be performed.	Cannot be performed.
Troubleshoot Fire Control System		
Primary: ASTS-1	Simulated by Exercise Number 8.	Simulated by Exercise Number 41
Primary: ASTS-2	Simulated by Exercise Number 69.	Cannot be performed.

helicopters would reflect differences in the missions performed by the aircraft, or differences in the special equipment on the aircraft. Similarly, the resident data for the Turret Mechanic's course for the M1A1 Tank should easily generalize to other tanks, and, with some changes, to other ground vehicles.

The three types of data are addressed in the resident data requirements include training device data, fidelity dimension data, and instructional feature data.

Fidelity Dimension Data

Fidelity dimensions and levels. In thinking about training and fidelity, the system that the students will use should be clearly distinguished from the environment in which that system will be used. For the flight trainer, the system is clearly the helicopter, and the environment is the terrain and enemy. Five visual variables and three variables related to terrain size and

visual special effects were used for the environment. Three variables related to motion and audio simulation addressed the system.

In a maintenance trainer for the M-1 tank, the system is not the tank, but rather the standard test equipment and breakout box. Standard test equipment and a breakout box are so basic and inexpensive that fidelity variations of these items within the OSBATS model are not likely to be productive. As shown in the last variable of Figure 10, this test equipment will either be available or not.

The M-1 tank is the students' training environment in the turret mechanic course. The key elements of this environment from a training standpoint are (a) the visual cues that are necessary for finding components and determining the best route of access for testing and replacement and (b) the actual physical representations required for using the test equipment while performing the task. Four variables address these elements using the major systems of the turret: (a) motion/hydraulics system, (b) electrical system, (c) fire control system, and (d) main weapon system. The hydraulics system has related motion requirements for the turret and gun tube so that the proper visual cues can be presented. In every case (except the main weapon system) the fidelity levels progress from none through very high quality replicas of components within each of the systems. Intermediate fidelity levels include two-dimensional panel trainer representations with pictures and controls (e.g., switches). Such a fidelity level for a given system would be useful for training the sequence of procedures but would provide no visual cues or practice with the test equipment. Troubleshooting tasks would not be trainable with this level of fidelity. An enhanced intermediate level of fidelity is termed interactive graphics: the presentation of high quality, two-dimensional visual cues as controlled by the student using a

		LEVEL							
	VARIABLE	1	2	3	4	5	6	7	8
1	Motion/Hydraulics	None	Picture and Contro	Interactive Graph	Electrical Movement	Hydraulic Traverse	Hydraulic Elevation	Hydraulic Controls	
2	Electrical System	None	Picture and Contro	Interactive Graph	Turret Networks B	Main Gun Safety Su	Gas Part Filter Sys	Ready Ammo Door	
3	Fire Control Sy	None	Picture and Contro	Interactive Graph	Laser Range Find	Commander's Sight	Computer Elect Unit	Hydraulic Turret Vlvirs	Accumulato
4	Main Weapon Sys	None	Breech Operating	CWS Firing Circuit					
5	Simulated Faults	None	Single Faults	Multiple Faults					
6	Auditory	None	Hydraulic	+ Solenoid Clicks	+ Moving Metal				
7	Test Equipment	None	STE-M1 and BOB						

Figure 10. Fidelity dimensions and levels for armor turret maintenance training.

computer-driven, video-disc based system, for example. Again no practice with the test equipment is provided.

Advanced levels of fidelity for all four system related variables include actual physical representations for portions of each system. These representations would be included in the model in the order of their contributions to the training of the tasks identified by the turret mechanic Program of Instruction. The hydraulic system includes the following levels beyond the two-dimensional representations: limited movement of the turret and gun tube via an electronic system so that the proper procedures and visual cues are possible from mock-ups of the hydraulic system; the actual hydraulic subsystems required for traversing, troubleshooting and maintaining the turret and then for elevating the gun tube; and finally realistic representations of the hydraulic system controls so that the students learn where and how the hydraulic system is activated.

Enhancements for the electrical system permit removal and replacement, troubleshooting and testing of the turret networks box, main gun safety switch, gas particulate filter system, and the ready ammunition door. Physical representations of the fire control system include the laser range finder, commander's sight, computer electronics unit, hydraulic turret valves, and accumulators. Elements of the main weapon system that would be evaluated include the breech operating mechanism and the firing circuit of the commander's weapon system.

The ability to simulate various faults within the M-1 turret system is analogous to the capability within a aviation training device to overlay various visual images to simulate points (e.g., cultural lights and moving vehicles) and areas (e.g., smoke and rotor wash). Levels of fidelity include individually simulating a set of faults such as a short in the turret networks box, followed by simulating several of the faults simultaneously.

Finally there are several auditory cues that can be represented to aid in troubleshooting and maintenance tasks: hydraulic system noises, solenoid clicks, and the rubbing and scraping of moving parts. Table 34 lists all seven of the fidelity dimensions and their levels that have been discussed.

Technical performance and cost data. Data concerning technical performance and development costs for these dimensions and levels of fidelity would be developed exactly the same as would be done for the aviation example. Subject matter experts

Table 34 Dimensions and Levels of Fidelity for a Turret Maintenance Trainer

Dimension 1 - Motion/Hydraulic System
- Level 1 - None
- Level 2 - Pictures and Controls on a Panel
- Level 3 - Automated, 2-dimensional, interactive graphics
- Level 4 - Limited movement via electronics and simulated hydraulics
- Level 5 - Actual hydraulics for traversing the turret
- Level 6 - Plus actual hydraulics for elevating the gun tube
- Level 7 - Plus actual hydraulic system controls

Dimension 2 - Electrical System
- Level 1 - None
- Level 2 - Pictures and Controls on a Panel
- Level 3 - Automated, 2-dimensional, interactive graphics
- Level 4 - Actual turret networks box
- Level 5 - Plus actual main gun safety switch
- Level 6 - Plus actual gas particulate filter system
- Level 7 - Plus actual ready ammunition door

Dimension 3 - Fire Control System
- Level 1 - None
- Level 2 - Pictures and Controls on a Panel
- Level 3 - Automated, 2-dimensional, interactive graphics
- Level 4 - Actual laser range finder
- Level 5 - Plus actual commander's sight
- Level 6 - Plus actual computer electronics system
- Level 7 - Plus actual hydraulic turret valves
- Level 8 - Plus actual accumulators

Dimension 4 - Main Weapon System
- Level 1 - None
- Level 2 - Actual breech operating mechanism
- Level 3 - Plus actual CWS firing circuit

Dimension 5 - Simulated Faults
- Level 1 - None
- Level 2 - Single Faults
- Level 3 - Multiple Faults

Dimension 6 - Auditory Cues
- Level 1 - None
- Level 2 - Hydraulic sounds
- Level 3 - Plus solenoid clicks
- Level 4 - Plus moving metal parts

Dimension 7 - Test Equipment
- Level 1 - None
- Level 2 - Actual STE-M1 and Break-out Box

(SMEs) would specify the technical performance indices for each level of each dimension. Then cost data from previous training devices or estimates from engineers would be used to develop statistical estimates for the parameters of the cost functions.

Fidelity rule base. The development of a fidelity rule base would require SME research, just as it did for the aviation example. We expect that it should be relatively easy to develop some of the rules in the rule base. For example, if a task requires removing a component, then we would expect that the task could not be trained on a training device unless the device represented that component so that it could be removed and replaced. However, rules for troubleshooting tasks are likely to be more complex, because of the extensive branching involved in these tasks.

Instructional Feature Data

Instructional features. In contrast to aviation tasks, maintenance tasks do not require a high degree of continuous control, nor do they possess significant time-sharing demands. Consequently, some of the instructional features appropriate for aviation would be excluded from armor maintenance, including (a) parameter freeze, (b) record/replay, and (c) remote replay.

Both trainers include a system freeze feature. This fact is consistent with the current instructional features rule base that states that system freeze is appropriate for long procedures. The turret trainer also includes a performance indicators feature. As presently implemented, this feature allows the instructor to encode the turret component that a student replaces. This feature is required because the system computer cannot detect the student's responses. In contrast, the panel trainer does not include this feature, presumably because the student's limited repertoire of permissible responses (button pushes, knob turns) are all computer detectable. This result is also consistent with the instructional feature rule base.

Of the 21 instructional features currently considered by the OSBATS model, the following instructional features are present in some form in one or both of the turret maintenance trainers.

1. Automated Performance Measurement
2. Performance Indicators
3. Procedure Monitoring
4. Total System Freeze
5. Initial Conditions
6. Scenario Control
7. Simulation Demonstration (manual)

Although these instructional features serve the same function as comparable ones in aviation training devices, their implementation may be significantly different in the current context from the implementation in aviation. Thus, one should take care when extrapolating data variables across domains.

The following instructional features from aviation training appear to be inappropriate in the armor maintenance context.

1. Simulator Record/Playback
2. Remote Graphics Replay
3. Crash Override
4. Parameter Freeze
5. Flight System Freeze
6. Position Freeze
7. Real-Time Simulation Variables Control

The remaining instructional features appear to be appropriate for some tasks, although they are not implemented in either existing trainer.

1. Automated Performance Alerts
2. Augmented Feedback
3. Augmented Cues
4. Reset/Reposition
5. Adjunct Computer-Assisted Instruction
6. Automated Adaptive Training
7. Automated Cueing and Coaching

We did not uncover additional instructional features that were not considered in aviation.

<u>Instructional feature cost</u>. We anticipate that instructional feature cost may be estimated using the same procedures as would be appropriate for aviation training. In this respect, we would also expect many of the same difficulties in determining instructional feature cost for armor training devices that have been experienced in assessing costs for aviation training devices. We do not think that it would be feasible to determine costs by extrapolating from aviation.

<u>Instructional feature benefit weight</u>. The instructional feature benefit weights were based on a survey of instructors regarding the extent to which they used various instructional features. We anticipate that this procedure would work in the armor maintenance domain, as well. However, since there are fewer training devices available, it would be more difficult to obtain as large a sample of respondents on which to base the feature weights. In addition, some features may not be implemented in any existing training device. For these features, the feature weight would be assessed by comparison to existing features, or by asking instructors if they would use such a feature if it existed.

<u>Instructional feature rules</u>. Many of the instructional feature rules appear to be the same in the armor domain as they are for aviation. We anticipate that a closer examination of how instructional features are used in armor maintenance training would reveal uses that were not considered in our analysis of aviation training. These new considerations could be represented

as additional instructional feature rules. We think it likely that most of the new rules would apply to aviation as well as armor.

Training Device Data

Training device data are descriptions of existing or hypothetical training devices according to cost, and according to the fidelity dimensions and instructional features defined above. The procedures for defining and collecting these data should be the same as those used in aviation training.

Task Data Requirements

Task data requirements include the data that are unique to each application of the OSBATS model. These data include task learning points, task simulation requirements, task training hours and costs, and data for the instructional feature and fidelity rule bases. For the most part, there does not appear to be any special problems in obtaining these data in the armor domain. Some comments on specific data categories follow.

1. Task learning points. The procedure used to scale entry level performance and the performance standard will probably be somewhat different for aviation maintenance tasks than for aviation operations tasks, because of the differences in the kinds of skills required. We anticipate that when scaling methods are defined, there will be several different kinds for continuous control, procedures, cognitive tasks, and so forth. Thus, the scaling procedure will probably depend more on skill type than training domain.

2. Simulation requirements. The reasons for simulated training are probably somewhat different in the current domain than they are in aviation. For example, damage to equipment is probably a somewhat greater concern here, while existence of training ranges is of no concern at all. This result would suggest the need to develop a comprehensive list of reasons for simulation, and a method of aggregating ratings of these reasons that is independent of domain.

3. Data for fidelity rule base. Since the rule base has not been developed, its data requirements are unknown. It seems likely that these requirements will be somewhat simpler than the requirements in aviation, because it is not necessary to specify the requirements for a visual system. Specification of the visual system accounts for most of the rules in the existing fidelity rule base.

4. Data for instructional feature rule base. It seems likely that the instructional feature rule base will be the same across domains. Hence, the data requirements will also be the same.

5. Training hours and costs. These data should be no different in this domain than they were in aviation.

Comparison of the Model Process

Our analysis did not uncover any changes in the general OSBATS model process that would be required to make it applicable for optimization of armor maintenance training systems. However, there are several differences in the two domains we have analyzed that have implications on the operation of the model.

The first difference is in the complexity. The aviation training example involved more tasks, more fidelity dimensions, and greater variety in the skills trained. This apparent difference is partly illusory, however. The turret mechanic must be able to perform a far greater number of tasks than are covered in the Program of Instruction. Many of these are listed as "related tasks," and are not trained under the assumption that the ability to perform these tasks will transfer from other tasks. If the related tasks were included in the analysis, then there would be potential implication on the OSBATS model, because of the possibility of transfer between tasks, which is not currently considered in the model.

A second difference is in the methods that must be used to determine fidelity requirements. The analysis of visual fidelity requirements in the fidelity rule base for aviation involves a detailed analysis of the kinds of activities required to perform a task, where there activities include such specific actions such as estimating altitude or range, or detecting distant targets. It seems that the analysis for the turret mechanic would not be as complex, and would not require the same depth of knowledge about the task. Thus, it might be more likely that the engineer who is the user of the OSBATS model might be able to provide the data for the fidelity rule base in maintenance problem, while considerably greater subject-matter knowledge would be required to provide comparable data for aviation maintenance.

A third difference is that the reasons for simulation are somewhat different in the two domains. This difference will have an impact on the kinds of factors that are considered in evaluating the benefits that may be derived from device-based training.

Summary and Conclusions

The overall conclusion of the analysis is that the OSBATS model should be directly applicable to the M-1 Abrams Turret Mechanic Course. No changes would be required in the general model processes and organization of modules. One specific change within the Simulation Configuration Module would likely be required. However, application of OSBATS to the new domain would require considerable development of resident data, particularly fidelity dimension data.

The difficulty of the modeling process revolves around the specification of the fidelity dimensions and levels. The complexity of the model is a function of both the system being used by the students and the environment in which the students use that system. For the flight trainer the system was the helicopter, which was considerably more complex than the test equipment of the maintenance trainer. However, the environment of the maintenance trainer was a complex tank turret, in some ways as complex than the environment of the aviation training device. The process of breaking the system and environment into dimensions and then defining levels of fidelity for each remains an art, but the process is much better understood now that it has been completed twice and, we believe, codifiable in the near future. What is required is feedback from maintenance trainers on the dimensions and levels described above and the development of dimensions and levels for at least one more application.

Once the fidelity dimensions and levels are specified, the model process proceeds systematically. The OSBATS data base must be developed around the definitions of these dimensions and levels. All of the data elements are defined; most of the data must be developed with the support of SMEs.

SUMMARY AND CONCLUSIONS

The high cost of training using actual weapon systems and the expanded capability of training technology have increased the potential value of simulation-based training. However, the complexity of weapon systems and their associated training systems has made the process of designing training devices much more difficult. The process of formulating a cost-effective training-device concept requires many tradeoff analyses that compare the cost and effectiveness of alternative design concepts.

The OSBATS model aids the system engineer responsible for formulating a training-device concept. The goal of the OSBATS model is to provide methods to produce training device designs that meet the training requirements at the minimum cost. The model helps the system engineer to design training devices and to incorporate them, along with actual equipment, in a training system.

This report has described the OSBATS model in detail from the viewpoint of both the model developer and the user. The model description has described the input data requirements, the specific model processes, and the products of the model analyses. In addition, the report has discussed the results of the formative evaluation activities conducted in this effort.

In this section of the report, we will summarize the capabilities of the OSBATS model, highlighting both the accomplishments made by this effort and the gaps in our knowledge that need to be addressed by future research. Our summary will focus on three issues. First, we will describe the analyses performed by the OSBATS model, paying particular attention to how these analyses may be used to meet specific needs of the system engineer. Second, we will highlight what we think are the major accomplishments of the development effort. Finally, we will summarize the activities that are required for validation and technology transfer.

Summary of OSBATS Functions

The OSBATS system provides tools to aid the tradeoff analyses required to design cost-effective training devices. The model allows design engineers to consider training effectiveness seriously when they develop a training-device design concept. It provides an interactive environment that allows the engineer to consider many more alternative designs than would be possible without the model. Using the OSBATS model, the user can perform comparative analyses that identify cost drivers, produce and evaluate alternative training-device design concepts, and specify cost-efficient ways to use training devices to meet the training requirements.

Five modules interact to help the engineer develop and evaluate training-device concepts. The engineer can use the

modules singly or in combination to address a wide variety of training-device design issues. The following list describes some of the analyses that can be performed using the OSBATS model.

1. Screen training requirements to determine which requirements can be met most appropriately using some kind of training device.

2. Identify tasks that can be trained adequately using a simple, inexpensive training device.

3. Compare the thousands of potential training-device design options to determine which ones meet the specific task training requirements at the lowest cost.

4. Examine the minimum fidelity levels required to train a task, based on the specific activities performed as a part of the task.

5. Determine which instructional support features are needed to maximize the efficiency with which the training requirements may be met on a specific training device.

6. Compare the cost effectiveness of training conducted using a sophisticated, full-mission simulator with training conducted using a combination of simpler, part-mission simulators.

7. Compare the cost-effectiveness of a design proposed by the user or other individual with a design of the same cost recommended by the model.

8. Determine how training time should be allocated among training devices and actual equipment.

9. Investigate the effect of limited availability of actual equipment or a training device on the training time and cost required to meet the training requirements.

Significant Accomplishments of This Research

Our three-year effort to develop the OSBATS model has produced several advancements in the state of the art for training-device optimization. These advancements build on the results of previous research and existing models. The following paragraphs summarize the most significant accomplishments, which distinguish the OSBATS model from predecessor models.

First, The OSBATS system provides a consistent approach to address a variety of training-device design problems. Its consistency comes from its top-down design, and its coordinated use of cost-benefit optimization in each of its components. Each module addresses one aspect of the training-system design process and recommends an optimal choice by considering the factors that affect the costs and benefits relevant to the process. The

modules share common concepts, such as learning rates, task weights, and cost elements, to ensure consistent results.

The design of the OSBATS system provides the flexibility required to accommodate the complex interactions involved in training-system design. This approach captures the inherently iterative nature of the training-system design process as described early in this report. The model provides methods that allow the results of analyses using any OSBATS module to provide information used by other modules. Thus, the model's modular structure allows easy analysis and refinement of results.

The characteristic of the OSBATS model that most distinguishes it from its predecessors is its emphasis on training-device design. The training-device design modules of the OSBATS model allow the user to investigate and compare many design options. The design engineer may use the results of these modules to determine which fidelity and instructional feature alternatives should be included in the training-device design, based on the training requirements. All other existing training-development models emphasize evaluation. Those models allow the user to evaluate a single training-device design. Application of other models to a large number of alternative designs would be overly burdensome on the design engineer. Thus, the OSBATS model has opened up a stage of the training-device design process to the benefits of analytic modeling.

The OSBATS model, unlike most others, aggregates cost and effectiveness estimates to develop recommendations based on a effectiveness/cost ratio. Other models apply a benefit analysis followed by a cost analysis. For example, Kribs, Simpson, and Mark (1983), in their review of medium selection models, identified five subtasks that were common to these models. These subtasks included a ranking of training media for training effectiveness followed by a cost tradeoff analysis used to perform the final selection. In a similar fashion, the instructional support feature guidelines developed for the Air Force (Logicon, 1985) specify a benefit analysis followed by an analysis of technology and cost considerations. The integrated effectiveness/cost analysis provided by OSBATS is superior to methods that perform sequential effectiveness and cost analyses, in that the latter methods tend to reject options that offer moderate benefit at a low cost in favor of options that offer high benefit at a high cost. Usually more overall effectiveness can be obtained within a cost budget with several moderately effective, but inexpensive options, than with one or two highly effective, but expensive options.

In developing the OSBATS model, we have produced the following other advancements in specific areas of training-system modeling.

1. We have extended the framework for training-device optimization initially proposed by Roscoe (1971) to consider the impact of constraints on the use of training device or

actual equipment on the time or cost required to meet training requirements.

2. We have developed new procedures to determine task fidelity and instructional feature requirements from descriptions of the activities involved in the tasks.

3. We have developed new methods to cluster tasks according to their needs for simulation and their requirements for a sophisticated simulation capability.

Needs for Future Model Development

The OSBATS model has been completely specified and prototype software has been developed. However, further work is needed before the model is transferred to the user community. The required activities include model expansion, data base development, model calibration and validation, and software enhancement. Some specific needs are outlined below.

Technology Transfer

The ultimate goal of the OSBATS research and development effort is the transfer of the software to the engineers responsible for training-device concept formulation. However, the current version of the system is not sufficiently developed to allow direct transfer to users. Barriers to technology transfer come from both limits in the state of model development, and from the process by which the model was developed.

The current OSBATS data base supports the use of the model over a limited domain. Although the specific domain of application is the AH-1 training course, we think that the model should be applicable with only minor changes to most training domains involving rotary-wing aviation operations. Use of the model outside of this domain requires the user either to collect additional data from subject-matter experts or to make assumptions about the values of such data and suffer a consequent loss in the accuracy of the model's predictions. Furthermore, operation of the model outside of the domain for which it was originally developed will probably require assistance from the model developer, the programmer, or both, to tailor the model to the new situation. Although our evaluation has indicated that the model processes are general, we expect that situations will arise in future applications that require modifications to the model or software.

Nevertheless, it should be stressed that it is possible, with appropriate assistance, to apply the model to a wide variety of problems. We think that the application of the model on an actual training design problem should be a high priority. The model application will establish a working relationship between model developers and model users. The feedback obtained from model users will provide a wealth of information that can be used to improve the model. In addition, we are confident that the

model will provide the engineer insights that can be used to produce a better training device concept.

The initial phase of the OSBATS development process was conducted with limited interactions with the eventual model users. The engineers were used primarily to evaluate the software and to provide information on the procedures currently used for training-device concept formulation. Future development should have much greater involvement by the engineers who will use the OSBATS model in concept formulation. We recommend that future development efforts include a mechanism that will provide an ongoing dialogue between the model developers and potential users to tailor the model to user needs, increase user ownership of the model, and support technology transfer.

Other needs for future model development support the need for technology transfer. That is, new model capabilities, more comprehensive data bases, easy data collection procedures, and model calibration and validation will all increase the likelihood of successful technology transfer, as well as offer other enhancements to the quality of the OSBATS model.

Additional Modeling Capabilities

We envision that as the OSBATS model is transferred to the training-device design engineers, many of the requirements for additional modeling capabilities will come from the user. At this stage in the development process, we have received some suggestions from potential users; other ideas have come from our own use of the model. The following list briefly describes several possible enhancements to the OSBATS model's capability.

1. Development of new task clustering methods that reflect other rationale for partitioning tasks, such as similarity of fidelity requirements, mission phase, and so forth. One of the critical early decisions in training-device design specifies the tasks used as the basis of the training-device design. The current Simulation Configuration Module contains one rationale for clustering tasks. Because of the importance of this decision, we think that a variety of task-clustering methods should be available to the user.

2. Enhancements to model integration capabilities. The OSBATS model currently includes several mechanisms that allow the results of one module to be used in a later module. Additional integration of the modules can improve their usefulness. For example, there is a need to incorporate the simulation requirements determined in the Simulation Configuration Module into the recommendations of the other modules.

3. Expansion of the model to new training technologies. New options that are available to the training-system designer should be evaluated as alternatives to traditional simulation-based methods. Examples of training methods that may require

more attention include embedded training, part-task training and skill training.

4. Develop ways to incorporate school requirements and constraints into the recommendations of the model. The school may require that certain features be included in a training-device design. Similarly, the school may have constraints on space or time that have an impact on the optimal training-device concept. There is a need for methods that allow all modules to consider these requirements in their analysis.

Data Collection Methods

One of the chief barriers to the application of the OSBATS model to a new training domain is the effort required to obtain the necessary data. There are several activities that could be accomplished to reduce the effort required for data collection. First, standard procedures for data collection should be developed. To the extent possible, these procedures should minimize the requirement for judgments by subject-matter experts. Where precise data are not available, methods for making assumptions about data values should be developed, and the impact of these assumptions on the results of the model determined. Finally, procedures should be developed to obtain required data from existing training data bases.

Model Calibration/Validation

The results of the OSBATS model hinge on several key assumptions about learning and transfer processes. Attempts to validate the model should focus on these key assumptions. The validation process will involve both determining the best value for key assumptions (calibration) and testing whether this assumption provides an adequate account of the learning and transfer processes addressed by the model (validation). Because of the large effort required for model validation, we recommend that the validation effort begin with a careful analytical evaluation of the model assumptions to determine which assumptions are the most critical to the model results. The sensitivity analyses conducted under the current effort should provide some guidance in identifying critical assumptions.

Software Enhancements

One of the requirements for technology transfer will be the development of production-quality software representing the OSBATS model. The production version of OSBATS will integrate analytic, rule-based, and data management capabilities of the model. We expect that the next version of OSBATS will incorporate several enhancements to the model software, such as a simplified user interface that is common to all modules, increased access to the logic that is used in rules bases, and access to the data that form the basis of the recommendations of the model. In addition, the next version of the software should

incorporate any new analytical and data management capabilities that are developed.

Some software enhancements may be investigated using the current prototype software. Candidates enhancements for development on the prototype system include user interface improvements, more sophisticated help capabilities, and additional displays. Development of these methods on the prototype software allows these methods to be evaluated before they are incorporated into the production software.

REFERENCES

Aagard, J. A., & Braby, R. (1976). Learning Guidelines and Algorithms for Types of Training Objectives, (TAEG Report No. 23). Orlando, FL: U.S. Navy Training Analysis and Evaluation Group.

Bickley, W.R., (1980). Training Device Effectiveness: Formulation and evaluation of a methodology. (Research Rep. 1291). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. (AD A122 777)

Card, S.K., Moran, T.P., & Newell, A. (1983). The psychology of human-computer interaction. Hillsdale, NJ: Erlbaum.

Carter, G., & Trollip, S. (1980). "A constrained maximization extension to incremental transfer effectiveness, or, how to mix your training technologies." Human Factors, 22(2), 141-152.

Cronholm, J.N. (1985). "The optimization of training systems", Proceedings of symposium on the transfer of training to military operational systems (pp. 237-259). Brussels, Belgium: NATO.

Department of the Army (1984). M1 organizational maintenance manual. TM 9-2350-255-20-2-3-1/2/3.

Department of the Army (1984). M1 organizational troubleshooting manual. TM 9-2350-255-20-2-2-1/2/3.

Department of the Army (1986). M1 turret organizational maintenance troubleshooting trainer, Device 17-86-1. TM 9-6910-243-10.

Department of the Army (1986). M1 tank turret organizational maintenance trainer, Device 17-67. TM 9-6910-242-10.

Department of the Army (1985). Soldier's Manual for MOS 45E, M1 Abrams tank turret mechanic. STP 9-45E12-SM-TG.

Elder, B.L., Gilligan, E.L., Sticha, P.J. (1988). User guide to OSBATS software (Version 1.02) (HumRRO Research Product RP-PRD-88-01). Alexandria, VA: Human Resources Research Organization.

Kribs, H.D., Simpson, A.C., & Mark, L.J. (1983). Automated instructional media selection (AIMS) (Tech. Rep. NAVTRAEQUIPCEN 79-C-0104-1). Orlando, FL: Naval Training Equipment Center.

Logicon, Inc. (1985). Instructional support feature guidelines (AF Contract No. F33615-84-C-0054). San Diego, CA: Author.

Newell, A., & Rosenbloom P. (1981). Mechanisms of skill acquisition and the law of practice. In J. R. Anderson (Ed.), Cognitive skills and their acquisition. Hillsdale, NJ: Erlbaum.

Pozella, D.J. (1983). Aircrew training devices: Utility and utilization of advanced instructional features Phase I--Tactical Air Command (AFHRL Tech. Rep. 83-22). Williams AFB, TX: Air Force Human Resources Laboratory.

Povenmire, H.K., & Roscoe, S.N. (1973). Incremental transfer effectiveness of a ground-based general aviation trainer. Human Factors, 15, 534-542.

Roscoe, S.N. (1971). Incremental transfer effectiveness. Human Factors, 15, 534-567.

Ross, D.T., & Schoman, K.E., Jr. (1977). Structured analysis for requirements definition. IEEE Transactions on Software Engineering. SE-3.

SofTech, Inc. (1981). Integrated computer-aided manufacturing (ICAM). Architecture Part II. Volume IV - Function modeling manual ($IDEF_0$). Waltham, MA: Author.

Sticha, P.J., Blacksten, H.R., Buede, D.M., & Cross, K.D. (1986). Optimization of simulation-based training systems. Volume III: Model description (HumRRO Final Report FR-PRD-86-13). Alexandria, VA: Human Resources Research Organization.

Sticha, P.J., Blacksten, H.R., Knerr, C.M., Morrison, J.E., & Cross, K.D. (1986). Optimization of simulation-based training systems. Volume II: Summary of the state-of-the-art (HumRRO Final Report FR-PRD-86-13). Alexandria, VA: Human Resources Research Organization.

Sticha, P.J., Singer, M.J., Blacksten, H.R., Mumaw, R. J., & Buede, D.M., (1987). Optimization of simulation-based training systems - Year II report (HumRRO Interim Report IR-PRD-87-30). Alexandria, VA: Human Resources Research Organization.

U.S. Army Ordnance Center and School (1987). Program of instruction for 643-45E10, M1 Abrams tank turret mechanic. POI 642-45E10.

Young, P.R., Luster, P.K., Stock, J.A. (PAR Government Systems Corporation), Mumaw, R.S., & Sticha, P.J. (HumRRO) (1986). Optimization of simulation-based training systems. Volume I: Plans for model implementation, evaluation and research (HumRRO Final Report FR-PRD-86-13). Alexandria, VA: Human Resources Research Organization.

APPENDIX A

OSBATS DATA BASE

I. Task Training Requirements

A. Task Learning Points

	ATM #	ABBREVIATED TASK TITLE	ENTRY	STD
	1000	VFR_FLIGHT	0.6900	0.70
*	1003	T/O_CHECKS	0.4941	0.70
*	1004	HOVER_PWR_CK	0.5417	0.70
*	1005	HOVER_FLT	0.4941	0.70
*	1006	NORMAL_T/O	0.4941	0.70
*	1007	S/MAX_P_T/O	0.4941	0.70
*	1008	DECEL_ACCEL	0.4941	0.70
*	1009	TRFC_PTRN	0.4941	0.70
	1010	FUEL_CK	0.6900	0.70
*	1012	DOPPLER_NAV	0.0000	0.70
*	1013	PRE_LNDG_CK	0.6837	0.70
*	1014	VMC_APPROACH	0.5000	0.70
*	1015	SHLW_APPR	0.4583	0.70
*	1016	CONFIND_OPN	0.5417	0.70
	1017	SLOPE_OPN	0.4167	0.70
*	1018	TERR_FLT_T/O	0.6250	0.70
*	1019	TERRAIN_FLT	0.6250	0.70
	1020	HVR_GRND_CK	0.6250	0.70
*	1021	NOE_DECEL	0.5417	0.70
*	1022	TRN_FLT_APP	0.5833	0.70
	1023	HI_SPD_FLT	0.5417	0.70
*	1024	HVR_AUTOROT	0.5417	0.70
	1025	ENG_FAIL_HVR	0.5417	0.70
*	1026	SIM_ENG_FAIL	0.5000	0.70
	1027	ENG_FAIL_SPD	0.4583	0.70
*	1028	MAN_THROTTLE	0.6250	0.70
*	1029	NO_SCAS/SAS	0.5000	0.70
	1033	ATT_RECVRY	0.3000	0.50
	1034	RADIO_PROC	0.6833	0.70
	1038	IFR_RECVRY	0.3000	0.50
*	1039	MSKG/UNMSKG	0.6917	0.70
	1040	TACT_COM_PRC	0.4941	0.70
-	1041	TRNSMT_RPT	0.5900	0.60
*	1042	MOVE_TECH	0.4900	0.50
	1043	ACTN_CONTACT	0.4900	0.50
	1044	WIRE_OBSTACL	0.6900	0.70
	1045	AFTER_LNDG	0.6917	0.70
*	1048	PIN/RDG_OPN	0.5417	0.70
	1049	FM_RAD_HOMNG	0.5833	0.7
	1053	TRNSMT_SIGNL	0.19	0.2
*	1054	OP_RMS	0.0814	0.7
*	1055	WPNS_PROC	0.0814	0.7
*	1056	OP_TURRET	0.0814	0.7
*	1057	OP_FFAR_ROC	0.0833	0.7
*	1058	OP_TOW_MSL	0.0814	0.7
	1059	ARM_SYS_MAL	0.0814	0.7

* indicates task included in sample problem

I. Task Training Requirements

A. Task Learning Points

	ATM #	ABBREVIATED TASK TITLE	ENTRY	STD
	1060	SAFE_WEAP	0.0814	0.7
*	1061	NOE_FIRING	0.0824	0.7
	1062	FIRNG_POSTN	0.0814	0.7
	1063	HI_SPD_DIVE	0.3333	0.7
*	2000	STD_AUTOROT	0.5417	0.7
	2001	LOW_AUTOROT	0.5417	0.7
	2002	HYDR_SYS_MAL	0.4583	0.7
*	2003	S/A_TRQ_MAL	0.3333	0.7
*	2004	L-A,H-S_AR	0.3333	0.7
	2005	LO-SPD_AUTOR	0.4167	0.7
	3000	EMER_START	0.407	0.7
	3004	CLUTCH_FL	0.4167	0.7
	3005	ENG_RESTART	0.4268	0.7
	3006	ENG_OVERSPD	0.4167	0.7
	3007	ENG_UNDERSPD	0.4167	0.7
*	3008	DROOP_CMP_FL	0.4167	0.7
	3009	ENG_CMPR_STL	0.4167	0.7
*	3010	TAIL_ROT_FL	0.3333	0.7
*	3011	MAST_BUMPING	0.6833	0.7
	3012	ENG_FIR_STRT	0.3333	0.7
	3013	FUSL_FIR_GUN	0.3333	0.7
	3014	FUSL_FIR_GRD	0.6837	0.7
	3015	FUSL_FIR_FLT	0.5833	0.7
	3016	ENG_FIR_FLT	0.5833	0.7
	3017	ENG_FR_L/ALT	0.5833	0.7
	3018	ENG_FR_CRUIS	0.5833	0.7
	3019	ELEC_FIR_FLT	0.5833	0.7
	3020	ELEC_FIR_GRD	0.5833	0.7
	3021	WING_STORES	0.5833	0.7
	3022	ECS_FUMES	0.6833	0.7
	3023	FUME_ELIMIN	0.6667	0.7
	3024	FUEL_BST_FL	0.6667	0.7
*	3025	THROTTLE_FL	0.5833	0.7
	3026	ELECTRIC_FL	0.5833	0.7
	3027	OVRHTD_BTTRY	0.6833	0.7
	3028	CAUTN_LGHT	0.6833	0.7
	3029	EMER_DESCENT	0.5833	0.7
*	3030	LNDG/TREES	0.5833	0.7
	3031	DTCH_PWR_ON	0.5833	0.7
	3032	DTCH_PWR_OFF	0.5833	0.7
*	3033	SCAS_FAILURE	0.5	0.7
	3034	BAILOUT_PRC	0.5	0.7
	3035	WNG_STR_JTSN	0.5833	0.7
	3036	TOW_EMER_PRC	0.4167	0.7
	3037	RUNAWAY_GUN	0.4167	0.7
	4003	ZERO_OIL_PR	0.6833	0.7
	4004	HI_EN_OIL_PR	0.6833	0.7
	4009	SCAS_PTCH_HO	0.5	0.7

APPENDIX A--OSBATS DATA BASE

I. Task Training Requirements

A. Task Learning Points

ATM #	ABBREVIATED TASK TITLE	ENTRY	STD
4010	SCAS_ROLL_HO	0.5	0.7
4011	SCAS_PTCH_MT	0.5	0.7
4012	SCAS_ROLL_MT	0.5	0.7
4013	SCAS_YAW_MT	0.5	0.7
4017	T/R_LOSS	0.29	0.3
4018	T/R_THRST_LS	0.29	0.3
4019	T/R_PTCH_FRZ	0.2	0.5
4020	GEAR_BX_LOSS	0.29	0.3
4022	FLAME_OUT	0.5	0.7
4028	ENG_CHP_LGHT	0.6753	0.7
4030	HI_TR_OIL_PR	0.6753	0.7
4031	HI_ENG_RPM	0.4167	0.7
4032	LO_ENG_RPM	0.4167	0.7
4035	CLOS_INL_GDV	0.5833	0.7
4036	OPEN_INL_GDV	0.69	0.7
4037	ENG_OIL_LOSS	0.6833	0.7
4038	TR_OIL_LOSS	0.6833	0.7
4039	TR_OIL_BYPAS	0.6833	0.7
4040	TR_CHIP_LGHT	0.6833	0.7
4042	ENG_DRV_FUEL	0.6833	0.7
4045	FORW_FUEL_LT	0.1628	0.7
4046	AFT_FUEL_LT	0.1628	0.7
4047	FUEL_FLTR_LT	0.6833	0.7
4048	LO_FUEL_LT	0.6833	0.7
4049	HYD/SYS_1_FL	0.0833	0.7
4050	HYD/SYS_2_FL	0.0833	0.7
4051	DUAL_HYDR_FL	0	0.4
4052	CMPLT_ELC_FL	0.5833	0.7
4055	D/C_GNRTR_FL	0.3333	0.7
4056	ALTERNATR_FL	0.3333	0.7
4057	A/C_INVRT_FL	0.4167	0.7

I. Task Training Requirements

B. Task Simulation Evaluation Factors

	ATM #	ABBREVIATED TASK TITLE	ABS RQRD	SPEC WX	SPEC SIT	SPEC EQUIP	TRNG EFFECT	TIME SAVE
	1000	VFR_FLIGHT	0	0	0	0	0	0
*	1003	T/O_CHECKS	0	0	0	0	0	1
*	1004	HOVER_PWR_CK	0	0	0	0	0	1
*	1005	HOVER_FLT	0	1	0	0	0	0
*	1006	NORMAL_T/O	0	1	0	0	0	1
*	1007	S/MAX_P_T/O	0	1	0	0	0	1
*	1008	DECEL_ACCEL	0	1	0	0	0	0
*	1009	TRFC_PTRN	0	1	0	0	0	0
	1010	FUEL_CK	0	0	0	0	0	0
*	1012	DOPPLER_NAV	0	0	0	0	0	0
*	1013	PRE_LNDG_CK	0	0	0	0	0	0
*	1014	VMC_APPROACH	0	1	0	0	0	1
*	1015	SHLW_APPR	0	1	0	0	0	1
*	1016	CONFIND_OPN	0	1	0	0	0	1
	1017	SLOPE_OPN	0	1	0	0	0	1
*	1018	TERR_FLT_T/O	0	1	0	0	0	1
*	1019	TERRAIN_FLT	0	1	0	0	0	1
	1020	HVR_GRND_CK	0	1	0	0	0	0
*	1021	NOE_DECEL	0	1	0	0	0	0
*	1022	TRN_FLT_APP	0	1	0	0	0	1
	1023	HI_SPD_FLT	0	0	0	0	0	0
*	1024	HVR_AUTOROT	0	0	0	0	0	1
	1025	ENG_FAIL_HVR	0	1	0	0	0	1
*	1026	SIM_ENG_FAIL	0	1	0	0	0	1
	1027	ENG_FAIL_SPD	0	1	0	0	0	1
*	1028	MAN_THROTTLE	0	1	0	0	0	1
*	1029	NO_SCAS/SAS	0	1	0	0	0	0
	1033	ATT_RECVRY	0	1	0	0	0	1
	1034	RADIO_PROC	0	0	0	0	0	0
	1038	IFR_RECVRY	0	1	0	0	0	1
*	1039	MSKG/UNMSKG	0	1	0	0	0	1
	1040	TACT_COM_PRC	0	0	1	0	0	1
	1041	TRNSMT_RPT	0	0	1	0	0	1
*	1042	MOVE_TECH	0	1	1	0	0	1
	1043	ACTN_CONTACT	0	1	1	0	0	1
	1044	WIRE_OBSTACL	0	0	0	1	0	1
	1045	AFTER_LNDG	0	0	0	0	0	0
*	1048	PIN/RDG_OPN	0	1	0	1	0	1
	1049	FM_RAD_HOMNG	0	1	0	0	0	1
	1053	TRNSMT_SIGNL	0	0	0	0	0	0
*	1054	OP_RMS	0	1	1	1	1	1
*	1055	WPNS_PROC	0	0	1	1	1	1
*	1056	OP_TURRET	0	1	1	1	1	1
*	1057	OP_FFAR_ROC	0	1	1	1	1	1
*	1058	OP_TOW_MSL	0	1	1	1	1	1
	1059	ARM_SYS_MAL	1	0	0	0	0	0
	1060	SAFE_WEAP	0	0	0	0	0	0
*	1061	NOE_FIRING	0	1	1	1	1	1

I. Task Training Requirements

B. Task Simulation Evaluation Factors

	ATM #	ABBREVIATED TASK TITLE	ABS RQRD	SPEC WX	SPEC SIT	SPEC EQUIP	TRNG EFFECT	TIME SAVE
	1062	FIRNG_POSTN	0	1	1	1	1	1
	1063	HI_SPD_DIVE	0	1	0	0	0	0
*	2000	STD_AUTOROT	0	1	0	0	1	1
	2001	LOW_AUTOROT	0	1	0	0	1	1
	2002	HYDR_SYS_MAL	0	1	0	0	1	1
*	2003	S/A_TRQ_MAL	0	1	0	0	1	1
*	2004	L-A,H-S_AR	0	1	0	0	1	1
	2005	LO-SPD_AUTOR	0	1	0	0	1	1
	3000	EMER_START	1	0	0	0	0	0
	3004	CLUTCH_FL	1	0	0	0	0	0
	3005	ENG_RESTART	1	0	0	0	0	0
	3006	ENG_OVERSPD	1	0	0	0	0	0
	3007	ENG_UNDERSPD	1	0	0	0	0	0
*	3008	DROOP_CMP_FL	1	0	0	0	0	0
	3009	ENG_CMPR_STL	1	0	0	0	0	0
*	3010	TAIL_ROT_FL	1	0	0	0	0	0
*	3011	MAST_BUMPING	1	0	0	0	0	0
	3012	ENG_FIR_STRT	1	0	0	0	0	0
	3013	FUSL_FIR_GUN	1	0	0	0	0	0
	3014	FUSL_FIR_GRD	1	0	0	0	0	0
	3015	FUSL_FIR_FLT	1	0	0	0	0	0
	3016	ENG_FIR_FLT	1	0	0	0	0	0
	3017	ENG_FR_L/ALT	1	0	0	0	0	0
	3018	ENG_FR_CRUIS	1	0	0	0	0	0
	3019	ELEC_FIR_FLT	1	0	0	0	0	0
	3020	ELEC_FIR_GRD	1	0	0	0	0	0
	3021	WING_STORES	1	0	0	0	0	0
	3022	ECS_FUMES	0	0	0	0	0	0
	3023	FUME_ELIMIN	0	0	0	0	0	0
	3024	FUEL_BST_FL	1	0	0	0	0	0
*	3025	THROTTLE_FL	1	0	0	0	0	0
	3026	ELECTRIC_FL	1	0	0	0	0	0
	3027	OVRHTD_BTTRY	1	0	0	0	0	0
	3028	CAUTN_LGHT	1	0	0	0	0	0
	3029	EMER_DESCENT	1	0	0	0	0	0
*	3030	LNDG/TREES	1	0	0	0	0	0
	3031	DTCH_PWR_ON	1	0	0	0	0	0
	3032	DTCH_PWR_OFF	1	0	0	0	0	0
*	3033	SCAS_FAILURE	0	0	0	0	1	1
	3034	BAILOUT_PRC	1	0	0	0	0	0
	3035	WNG_STR_JTSN	1	0	0	0	0	0
	3036	TOW_EMER_PRC	1	0	0	0	0	0
	3037	RUNAWAY_GUN	1	0	0	0	0	0
	4003	ZERO_OIL_PR	1	0	0	0	0	0
	4004	HI_EN_OIL_PR	1	0	0	0	0	0
	4009	SCAS_PTCH_HO	1	0	0	0	0	0
	4010	SCAS_ROLL_HO	1	0	0	0	0	0
	4011	SCAS_PTCH_MT	1	0	0	0	0	0

APPENDIX A--OSBATS DATA BASE

I. Task Training Requirements

B. Task Simulation Evaluation Factors

ATM #	ABBREVIATED TASK TITLE	ABS RQRD	SPEC WX	SPEC SIT	SPEC EQUIP	TRNG EFFECT	TIME SAVE
4012	SCAS_ROLL_MT	1	0	0	0	0	0
4013	SCAS_YAW_MT	1	0	0	0	0	0
4017	T/R_LOSS	1	0	0	0	0	0
4018	T/R_THRST_LS	1	0	0	0	0	0
4019	T/R_PTCH_FRZ	0	0	0	0	0	1
4020	GEAR_BX_LOSS	1	0	0	0	0	0
4022	FLAME_OUT	1	0	0	0	0	0
4028	ENG_CHP_LGHT	1	0	0	0	0	0
4030	HI_TR_OIL_PR	1	0	0	0	0	0
4031	HI_ENG_RPM	1	0	0	0	0	0
4032	LO_ENG_RPM	1	0	0	0	0	0
4035	CLOS_INL_GDV	1	0	0	0	0	0
4036	OPEN_INL_GDV	1	0	0	0	0	0
4037	ENG_OIL_LOSS	1	0	0	0	0	0
4038	TR_OIL_LOSS	1	0	0	0	0	0
4039	TR_OIL_BYPAS	1	0	0	0	0	0
4040	TR_CHIP_LGHT	1	0	0	0	0	0
4042	ENG_DRV_FUEL	1	0	0	0	0	0
4045	FORW_FUEL_LT	1	0	0	0	0	0
4046	AFT_FUEL_LT	1	0	0	0	0	0
4047	FUEL_FLTR_LT	1	0	0	0	0	0
4048	LO_FUEL_LT	1	0	0	0	0	0
4049	HYD/SYS_1_FL	1	0	0	0	0	0
4050	HYD/SYS_2_FL	1	0	0	0	0	0
4051	DUAL_HYDR_FL	1	0	0	0	0	0
4052	CMPLT_ELC_FL	1	0	0	0	0	0
4055	D/C_GNRTR_FL	1	0	0	0	0	0
4056	ALTERNATR_FL	1	0	0	0	0	0
4057	A/C_INVRT_FL	1	0	0	0	0	0

II. Other Task Data

A. Task Training Hours and Costs

	ATM #	ABBREVIATED TASK TITLE	CLASSRM HOURS	---- EQUIPMENT HOURS ---- NON-FLT	FLT	SETUP	OTHER EQUIP ($000)
	1000	VFR_FLIGHT	0.1	0.00	0.00	0	0
*	1003	T/O_CHECKS	2	0.70	0.00	0	0
*	1004	HOVER_PWR_CK	0.2	0.00	0.20	0	0
*	1005	HOVER_FLT	0	0.00	0.40	0	0
*	1006	NORMAL_T/O	0	0.10	0.40	0.1	0
*	1007	S/MAX_P_T/O	0	0.10	0.40	0.1	0
*	1008	DECEL_ACCEL	0	0.00	0.60	0	0
*	1009	TRFC_PTRN	0	0.10	0.50	0	0
	1010	FUEL_CK	0	0.10	0.20	0	0
*	1012	DOPPLER_NAV	6	0.30	0.70	0	0
*	1013	PRE_LNDG_CK	0.2	0.00	0.10	0	0
*	1014	VMC_APPROACH	0	0.10	0.60	0.1	0
*	1015	SHLW_APPR	0	0.10	0.60	0.1	0
*	1016	CONFIND_OPN	0	0.10	0.60	0.1	0
	1017	SLOPE_OPN	0	0.10	0.30	0	0
*	1018	TERR_FLT_T/O	0	0.00	0.10	0	0
*	1019	TERRAIN_FLT	0	0.00	0.40	0	0
	1020	HVR_GRND_CK	0.2	0.10	0.40	0	0
*	1021	NOE_DECEL	0	0.00	0.30	0	0
*	1022	TRN_FLT_APP	0	0.00	0.20	0	0
	1023	HI_SPD_FLT	0.1	0.10	0.10	0	0
*	1024	HVR_AUTOROT	0	0.10	0.30	0.1	0
	1025	ENG_FAIL_HVR	0	0.10	0.30	0	0
*	1026	SIM_ENG_FAIL	0	0.10	0.20	0	0
	1027	ENG_FAIL_SPD	0	0.10	0.10	0	0
*	1028	MAN_THROTTLE	0.5	0.10	0.60	0.1	0
*	1029	NO_SCAS/SAS	0.1	0.10	0.50	0	0
	1033	ATT_RECVRY	0	0.20	0.10	0	0
	1034	RADIO_PROC	0	0.20	0.10	0	0
	1038	IFR_RECVRY	0	0.20	0.30	0.1	0
*	1039	MSKG/UNMSKG	0	0.00	0.40	0.1	0
	1040	TACT_COM_PRC	0.2	0.00	0.00	0	0
	1041	TRNSMT_RPT	0.2	0.00	0.00	0	0
*	1042	MOVE_TECH	0.4	0.30	0.00	0	0
	1043	ACTN_CONTACT	0.3	0.00	0.00	0	0
	1044	WIRE_OBSTACL	0	0.10	0.00	0	0
	1045	AFTER_LNDG	0.1	0.10	0.10	0	0
*	1048	PTN/RDG_OPN	0.3	0.00	0.60	0.1	0
	1049	FM_RAD_HOMNG	0.3	0.1	0.3	0.1	0
	1053	TRNSMT_SIGNL	0.3	0.1	0	0	0
*	1054	OP_RMS	5	0.2	1.5	0.5	1.39
*	1055	WPNS_PROC	3	0.3	2	0.6	1 87
*	1056	OP_TURRET	4	0.1	2	0.6	0.3
*	1057	OP_FFAR_ROC	0.1	0.1	2	0.6	4.98
*	1058	OP_TOW_MSL	9	0.3	2	0.6	0.5
	1059	ARM_SYS_MAL	0.5	0.3	0	0	0
	1060	SAFE_WEAP	0.3	0.3	0.5	0	0
*	1061	NOE_FIRING	0.2	0.3	3.5	0.6	8.87

II. Other Task Data

A. Task Training Hours and Costs

	ATM #	ABBREVIATED TASK TITLE	CLASSRM HOURS	EQUIPMENT HOURS NON-FLT	EQUIPMENT HOURS FLT	EQUIPMENT HOURS SETUP	OTHER EQUIP ($000)
	1062	FIRNG_POSTN	0	0.4	0.5	0	0
	1063	HI_SPD_DIVE	0	0.2	0.5	0.1	0
*	2000	STD_AUTOROT	0	0.1	0.7	0.2	0
	2001	LOW_AUTOROT	0	0.2	0.7	0.2	0
	2002	HYDR_SYS_MAL	0	0.2	0.7	0.2	0
*	2003	S/A_TRQ_MAL	0	0.1	0.7	0.2	0
*	2004	L-A,H-S_AR	0	0.1	0.7	0.2	0
	2005	LO-SPD_AUTOR	0	0.1	0.7	0.2	0
	3000	EMER_START	0.2	0.1	0	0	0
	3004	CLUTCH_FL	0.3	0.1	0	0	0
	3005	ENG_RESTART	0.2	0.1	0	0	0
	3006	ENG_OVERSPD	0.2	0.1	0	0	0
	3007	ENG_UNDERSPD	0.2	0.1	0	0	0
*	3008	DROOP_CMP_FL	0.2	0.3	0	0	0
	3009	ENG_CMPR_STL	0.3	0.1	0	0	0
*	3010	TAIL_ROT_FL	0	0.1	0	0	0
*	3011	MAST_BUMPING	0.2	0.1	0	0	0
	3012	ENG_FIR_STRT	0.2	0.1	0	0	0
	3013	FUSL_FIR_GUN	0.1	0.1	0	0	0
	3014	FUSL_FIR_GRD	0.1	0.1	0	0	0
	3015	FUSL_FIR_FLT	0.1	0.1	0	0	0
	3016	ENG_FIR_FLT	0.1	0.1	0	0	0
	3017	ENG_FR_L/ALT	0.1	0.1	0	0	0
	3018	ENG_FR_CRUIS	0.1	0.1	0	0	0
	3019	ELEC_FIR_FLT	0.2	0.1	0	0	0
	3020	ELEC_FIR_GRD	0.1	0.1	0	0	0
	3021	WING_STORES	0.2	0.1	0	0	0
	3022	ECS_FUMES	0.2	0.1	0	0	0
	3023	FUME_ELIMIN	0.2	0.1	0	0	0
	3024	FUEL_BST_FL	0.3	0.2	0	0	0
*	3025	THROTTLE_FL	0.1	0.1	0	0	0
	3026	ELECTRIC_FL	0.1	0.1	0	0	0
	3027	OVRHTD_BTTRY	0.2	0.1	0	0	0
	3028	CAUTN_LGHT	0.1	0.1	0	0	0
	3029	EMER_DESCENT	0	0.1	0	0	0
*	3030	LNDG/TREES	0	0.2	0	0	0
	3031	DTCH_PWR_ON	0	0.2	0	0	0
	3032	DTCH_PWR_OFF	0	0.2	0	0	0
*	3033	SCAS_FAILURE	1.2	0.1	0.2	0	0
	3034	BAILOUT_PRC	0	0.2	0	0	0
	3035	WNG_STR_JTSN	0.2	0.1	0	0	0
	3036	TOW_EMER_PRC	0.3	0.3	0	0	0
	3037	RUNAWAY_GUN	0.2	0.2	0	0	0
	4003	ZERO_OIL_PR	0.1	0.1	0	0	0
	4004	HI_EN_OIL_PR	0.1	0.1	0	0	0
	4009	SCAS_PTCH_HO	0.2	0.2	0	0	0
	4010	SCAS_ROLL_HO	0.2	0.2	0	0	0
	4011	SCAS_PTCH_MT	0.2	0.2	0	0	0

II. Other Task Data

A. Task Training Hours and Costs

ATM #	ABBREVIATED TASK TITLE	CLASSRM HOURS	EQUIPMENT HOURS NON-FLT	FLT	SETUP	OTHER EQUIP ($000)
4012	SCAS_ROLL_MT	0.2	0.2	0	0	0
4013	SCAS_YAW_MT	0.2	0.2	0	0	0
4017	T/R_LOSS	0.2	0.1	0	0	0
4018	T/R_THRST_LS	0.2	0.1	0	0	0
4019	T/R_PTCH_FRZ	0.2	0.1	0	0	0
4020	GEAR_BX_LOSS	0.2	0.2	0	0	0
4022	FLAME_OUT	0.2	0.1	0	0	0
4028	ENG_CHP_LGHT	0.2	0.1	0	0	0
4030	HI_TR_OIL_PR	0.1	0.1	0	0	0
4031	HI_ENG_RPM	0.2	0.1	0	0	0
4032	LO_ENG_RPM	0.2	0.1	0	0	0
4035	CLOS_INL_GDV	0.2	0.1	0	0	0
4036	OPEN_INL_GDV	0.2	0.1	0	0	0
4037	ENG_OIL_LOSS	0.1	0.1	0	0	0
4038	TR_OIL_LOSS	0.1	0.1	0	0	0
4039	TR_OIL_BYPAS	0.2	0.1	0	0	0
4040	TR_CHIP_LGHT	0.2	0.1	0	0	0
4042	ENG_DRV_FUEL	0.3	0.1	0	0	0
4045	FORW_FUEL_LT	0.2	0.1	0	0	0
4046	AFT_FUEL_LT	0.2	0.1	0	0	0
4047	FUEL_FLTR_LT	0.2	0.1	0	0	0
4048	LO_FUEL_LT	0.2	0.1	0	0	0
4049	HYD/SYS_1_FL	0.5	0.2	0.3	0.1	0
4050	HYD/SYS_2_FL	0.5	0.2	0.3	0.1	0
4051	DUAL_HYDR_FL	0.5	0.2	0.3	0.1	0
4052	CMPLT_ELC_FL	0.2	0.1	0	0	0
4055	D/C_GNRTR_FL	0.2	0.1	0	0	0
4056	ALTERNATR_FL	0.2	0.1	0	0	0
4057	A/C_INVRT_FL	0.2	0.1	0	0	0

III. Training Device Data

A. Training Device Costs

MEDIUM	COST (K) INVEST	COST (K) FIXED/YR	COST (K) VAR/HR	LIFE-CYCLE	UTIL (HR/YR)
CLASSROOM	0.0	5.0	0.0299	50	2000
COCKPIT PROC TRNR	80.0	2.0	0.0500	10	2000
DOPPLER NAV TRNR	40.0	1.0	0.0305	10	2000
IMC FLIGHT TRNR	3600.0	8.0	0.0700	10	3000
VMC FLIGHT TRNR	8000.0	40.0	0.0800	10	3000
WEAPONS TRNR	640.0	8.0	0.0500	10	2000
NOE NAV TRNR	10.0	0.4	0.0300	10	2000
ACT EQUIP(HLCPTR)	5000.0	100.0	0.6250	10	2500

III. Training Device Data

B. Training Device Cue and Response Capabilities

MEDIUM	VISUAL							MOTION			
	RES	CONT	TXTR	FRONT FOV	SIDE FOV	F/X PNTS	F/X AREA	PLTFM	SEAT	SOUND F/X	MAP AREA
CLASSROOM	0	0	0	0	0	0	0	0	0	0	0
COCKPIT PROC TRNR	0	0	0	0	0	0	0	0	0	0.27	0
DOPPLER NAV TRNR	0	0	0	0	0	0	0	0	0	0.27	0
IMC FLIGHT TRNR	0	0	0	0	0	0	0	0.63	0.36	0.27	0.58
VMC FLIGHT TRNR	0.8	0.58	0.5	0.86	0.51	0.18	0.45	0.9	0.36	0.54	0.62
WEAPONS TRNR	0.93	0.9	0.9	0.77	0	0.9	0.45	0	0	0.54	0.82
NOE NAV TRNR	0.93	0.9	0.9	0.86	0.51	0	0	0	0	0.27	0.62
ACTUAL EQUIPMENT	1	1	1	1	1	1	1	1	1	1	1

III. Training Device Data

C. Training Device Instructional Features

MEDIUM	PERF MEAS	PERF IND	PROC MNTR	PERF ALRT	AUG FDBK	AUG CUES	RECRD PLBK	SYSTM FREEZ	GRPH RPLY	INIT COND	SCEN CNTL	CRSH OVRD	RE-SET	PARM FREEZ
CLASSROOM	0	0	0	0	0	1	0	0	0	1	0	0	0	0
COCKPIT PROC TRNR	1	0	0	0	0	0	0	0	0	1	0	0	0	0
DOPPLER NAV TRNR	1	0	1	0	0	0	0	0	0	1	1	0	0	0
IMC FLIGHT TRNR	1	0	1	0	0	0	1	1	0	1	1	1	1	0
VMC FLIGHT TRNR	1	0	1	0	1	1	1	1	0	1	1	1	1	1
WEAPONS TRNR	1	0	0	0	1	0	0	0	0	1	1	0	0	0
NOE NAV TRNR	1	1	0	0	0	1	0	0	0	1	1	0	0	0
ACTUAL EQUIPMENT	0	0	0	0	0	0	0	0	0	0	0	0	0	0

III. Training Devic

C. Training

MEDIUM	FLT FREEZ	PSTN FREEZ	REAL TIME	AUTO DEMO	ADJN CAI	ADPT TRNG	CUE COACH
CLASSROOM	0	0	0	0	0	0	0
COCKPIT PROC TRNR	0	0	0	0	0	0	0
DOPPLER NAV TRNR	0	0	0	0	0	0	0
IMC FLIGHT TRNR	0	0	0	1	0	0	0
VMC FLIGHT TRNR	1	1	1	1	0	0	0
WEAPONS TRNR	0	0	0	1	0	0	0
NOE NAV TRNR	0	0	0	0	0	0	0
ACTUAL EQUIPMENT	0	0	0	0	0	0	0

IV. Fidelity Dimension Data

A. Fidelity Dimensions and Levels

DIMENSION	LEVEL	NAME	TECH PERF
VIS RESOLTN	1	M2 @ 0.3 KM	0.52
	2	M2 @ 0.5 KM	0.71
	3	M2 @ 1.0 KM	0.86
	4	M2 @ 2.0 KM	0.93
	5	M2 @ 3.0 KM	0.95
	6	M2 @ 4.0 KM	0.96
VIS CONTENT	1	PLANE WITH TREES	0.5
	2	ADD GENERIC FEATRS	0.58
	3	REALISTIC DENSITY	0.66
	4	LOW DNSTY HYDROGRAPH	0.74
	5	MED DNSTY HYDROGRAPH	0.82
	6	HI DNSTY HYDROGRAPH	0.9
VIS TEXTURE	1	LINES + POLYGONS	0.5
	2	MODULATING FNCTNS	0.62
	3	FEW DIGITIZED FOTOS	0.74
	4	MORE DIGITIZED FOTOS	0.84
	5	MANY DIGITIZED FOTOS	0.9
FRONT FOV	1	40 X 40 DEG	0.77
	2	40 X 50 DEG	0.86
	3	40 X 60 DEG	0.94
SIDE FOV	1	1 - 40 X 40 DEG	0.46
	2	1 - 40 X 50 DEG	0.51
	3	1 - 50 X 50 DEG	0.57
	4	1 - 50 X 60 DEG	0.62
	5	2 - 40 X 50 DEG	0.72
	6	2 - 40 X 60 DEG	0.79
	7	2 - 50 X 60 DEG	0.88
SPL F/X PNTS	1	NONE	0
	2	CULTURAL LGHTS	0.09
	3	ADD WPNS BLAST	0.18
	4	ADD DAMGD VEHICLES	0.36
	5	ADD AIRBRN VEHICLES	0.63
	6	ADD MVNG GND VEHCLE	0.9
SPL F/X AREA	1	NONE	0
	2	SMOKE AND DUST	0.45
	3	ROTOR WASH	0.9
MTN PLTFRM	1	NONE	0
	2	3 D.F.	0.36
	3	5 D.F.	0.63
	4	6 D.F.	0.9
MTN SEAT	1	STATIONARY	0
	2	SEAT SHAKER	0.36
	3	ADD G-SEAT	0.9
AUDIO F/X	1	NONE	0
	2	WPNS,SKID,SOME FAILS	0.27
	3	ADD NORMAL OPS NOISE	0.54
	4	ADD ABNORMAL OPS NSE	0.9

IV. Fidelity Dimension Data

A. Fidelity Dimensions and Levels

DIMENSION	LEVEL	NAME	TECH PERF
DATA SZ/CAL	1	5 X 5 KM	0.5
	2	10 X 10 KM	0.58
	3	10 X 20 KM	0.62
	4	10 X 30 KM	0.66
	5	20 X 30 KM	0.74
	6	30 X 30 KM	0.82
	7	30 X 40 KM	0.9

V. Instructional Feature Data

B. Instructional Feature Cost and Weight

FEATURE	BENEFIT WEIGHT	COST ($K)
Automated Performance Measurement	1.4	215.00
Performance Indicators	1.4	51.00
Procedure Monitoring	3.4	74.00
Automated Performance Alerts	1.3	61.00
Augmented Feedback	1.3	97.00
Augmented Cues	1.3	155.00
Record/Playback	1.42	92.00
Total System Freeze	3.1	24.00
Remote Graphics Replay	2.7	76.00
Initial Conditions	4.3	98.00
Scenario Control	4.4	155.00
Crash Override	4.77	29.00
Reset/Reposition	4.63	30.00
Parameter Freeze	2.95	35.00
Flight System Freeze	3.8	35
Positional Freeze	3.3	35
Real-Time Simulation Variables Control	3.7	137
Automated Simulator Demonstration	2.16	54
Adjunct CAI	1.7	236
Automated Adaptive Training	1.59	253
Automated Cueing and Coaching	1.1	156

VI. Training System Data

A. Course and System Information

Variable	Value
REQUIRED GRADS/YR	800

VI. Training System Data

B. Model Information

VARIABLE	VALUE
STANDARD ADJUSTMENT RATE	30%
LEARNING CURVE EXPONENT	0.7
MAXIMUM INST FEATURE EFFECT	10%
MAX NUMBER OF INST FEATURES	3
SETUP SAVINGS %	60%
COST SAVINGS WEIGHT	0.5
RECOMMENDATION BOUNDARY	0.5
ANNUAL FIXED COST FACTOR	300
HRLY VARIABLE COST FACTOR	80,000
NEW DEVICE UTILIZATION (HR)	3000
NEW DEVICE LIFECYCLE (YR)	10

APPENDIX B

FOUNDATIONS AND DERIVATIONS OF OSBATS MODEL CONSTRUCTS

Appendix B
FOUNDATIONS AND DERIVATIONS OF OSBATS MODEL CONSTRUCTS

Introduction

The essential OSBATS goal is to specify training systems that train students to required performance levels at minimum cost. To tackle this problem it is necessary to develop suitable (1) training-device learning curves and transfer functions, (2) cost functions, and (3) optimization techniques. This appendix provides supplementary details of the OSBATS model constructs in these three areas.

For convenience of dissertation, the simplified mathematical notation used during model development is employed in this appendix. This is, for the most part, different from that in the body of the report. Also, some variables are defined differently depending on context. A glossary, provided at the end of this appendix, defines variables and parameters and identifies the long form notation used in the main report.

References are also found at the end of the appendix.

Medium Effectiveness: Learning Curve and Transfer Functions

Summary

There are a number of goals in learning curve and transfer function selection and development. First, a learning curve must be able to represent actual student performance increase with reasonable accuracy, as must a transfer function represent performance transfer to actual equipment. Second, it must be possible to develop procedures to relate learning curve and transfer function parameters to training system design decision variables. Third, learning curves and transfer functions must be

efficiently calculable, since they will be evaluated a great many times in the various OSBATS modules. Fourth, learning curves should have closed form inverse functions, since these are used in other methodology development; these learning curve inverse functions should also be efficiently calculable.

One simplifying assumption is made at this point, namely, that the performance scales for the various devices and actual equipment are defined in such a way that transfer functions from training devices to actual equipment are simple linear, i.e., that the performance score on actual equipment will be the performance score on the training device multiplied by a transfer ratio.

A methodology is then required for determining medium (training device) effectiveness, i.e., for (1) constructing the learning curve $P_d(t)$ representing the learning on the device as measured on a normalized (0-1) scale, and (2) determining the transfer ratio a_r from that device to the actual equipment, so that a learning curve $P(t) = a_r P_d(t)$ will represent the learning on the training device in terms of the performance a student would achieve on the actual equipment after training only on the training device. What seems particularly appealing is the decomposition of the process in such a way that P(t) is a function $P(t;s,m,a_r,c,k)$ of parameters s, m, a_r, c, and k, which may be determined sequentially (inverse order) as follow:

(1) k is a learning curve shape parameter, viz., the exponent in a power function form, and will be the same as that for the actual equipment; k may either be based on the literature (e.g. Card, Moran, and Newell, 1983) or on a fit to empirical learning data for actual equipment.

(2) c is another learning curve shape parameter, and serves, further, as a scaling constant used to convert time units to dimensionless units; c will be the same

as for actual equipment and will be based on a fit to empirical learning data for actual equipment.

(3) a_r, the transfer ratio, will be a function of the medium fidelity levels $\{r_i\}$ in the different fidelity dimensions, indexed by i, together with SME estimates of the target fidelity levels $\{R_i\}$ for a training device to train the task. If all fidelity levels are met exactly then the task can be trained to a level $Y+x(1-Y)$ exceeding the standard Y by a specified amount, where the fraction x is a constant. Prior to the determination of a_r, a target transfer ratio a_R is determined according to the $\{R_i\}$ definition chosen.) The algorithm for a_R also may involve constants $\{f_i\}$, giving for each fidelity dimension the fractional degradation in transfer that would occur for any task for which that fidelity is applicable, as that fidelity level is decreased from 1 to 0.

(4) m is a time-multiplier used to account for increased time efficiency engendered by instructional features and reduced setup time.

(5) s is an effective head start, chosen so that the entry performance P(0) equals a specified value.

The determination process is represented in Figure B-1.

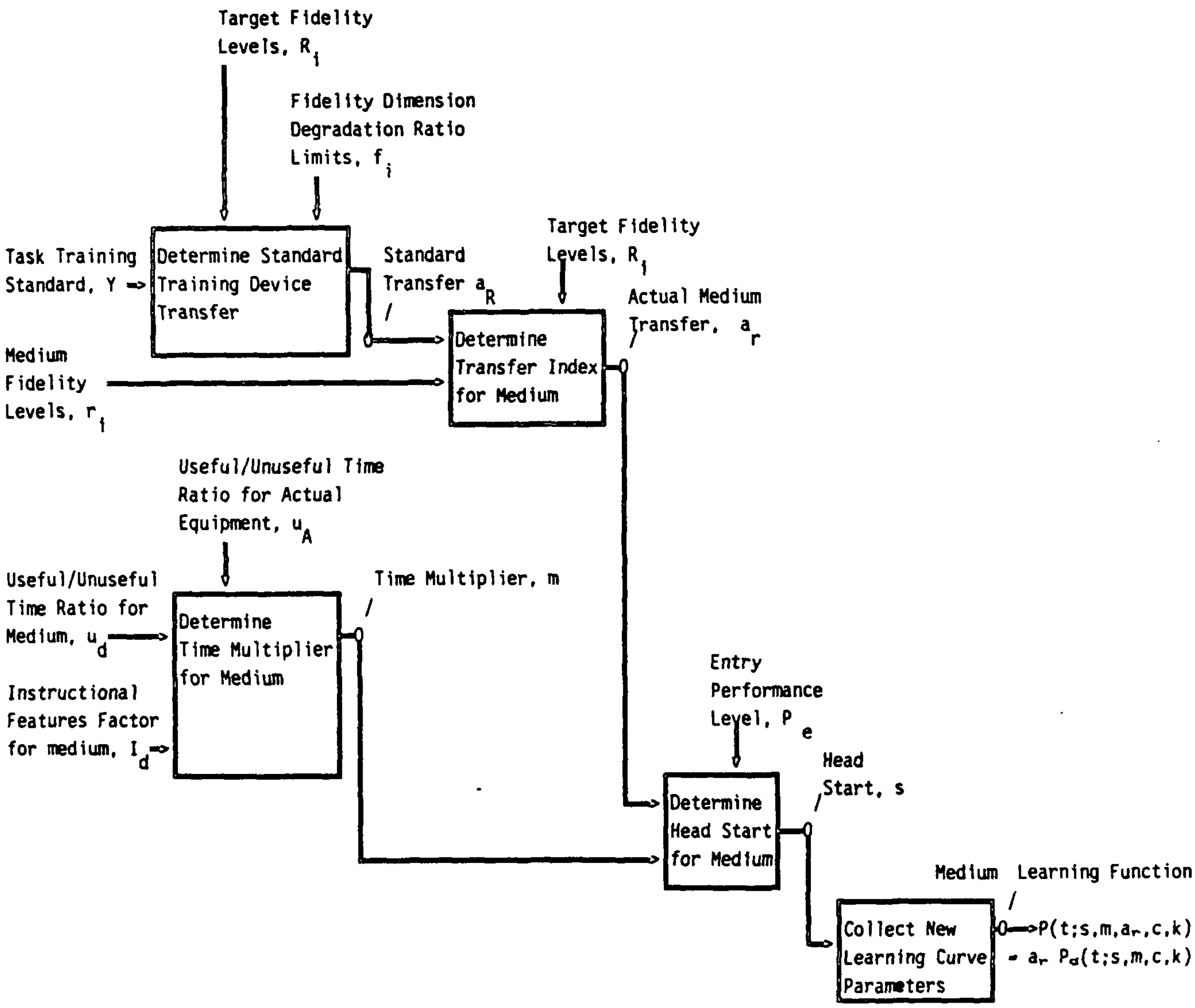


Figure B-1. Medium effectiveness (learning curve and transfer ratio) determination

Scale Invariance

Dimensional analysis considerations require the use of a scaling constant for time variables. This is basically the parameter c discussed above, though it may also include a factor to control the shape of the curve. Failing to introduce such a scaling factor would force the incorporation of messy, iterative, and basically unappealing procedures when tackling the medium effectiveness estimation problem. Without c the learning curve cannot be adjusted for a change in time scale. If there is a coefficient c preceding (s+t) in the learning curve, then whether we measure time in minutes or hours makes no difference -- provided c is adjusted accordingly. The curve will be the same as before in the sense that it will take the same length of time t_2-t_1 to go from performance level y_1 to y_2 no matter how time is measured. This scale invariance also allows the simplified incorporation of instructional features and time usage efficiency in the medium learning curves.

Formally, a learning curve is scale invariant if it has the form f({c[s+t]}, and other parameters). I.e., c, s, and t always appear together in the equation in this form c[s+t]. The other parameters may include such things as the shape constant k. Each of the following is scale invariant, by this definition:

Negative Exponential:

$$P(t) = 1 - e^{-c(s+t)}$$

Hyperbolic:

$$P(t) = \frac{c(s+t)}{c(s+t) + 1}$$

In the same way the learning curves associated with virtually any of the standard learning theories (replacement, accumulation, strength (Hull), all-or-none) can be written in a scale invariant form, as can, too, most empirical fits (e.g., logistic).

Power Functions

Despite the popularity of the learning curve forms identified in the last paragraph, Newell and Rosenbloom (1981) have shown that a multitude of learning achievement measures follow a power function curve. Their strong empirical evidence and supporting rationale motivated our selection of a power function form for the OSBATS learning curves (Sticha, et.al., (1986)). Actually, Newell and Rosenbloom used time-to-perform on a trial, errors to perform on a trial, and linear combinations of these as the measures that, when plotted on a log-log plot against time proved to be so nicely linear, e.g., fitting the form:

$$\log(T) = -k \ln(s+t) + \ln(B)$$

where T represents the performance measure, and where k and B are positive constants. But then

$$T = B\ (s+t)^{-k} \qquad (1)$$

from whence the term power function rule[1]. Note that this may be written

$$T = \left[B^{-1/k}\ (s+t)\right]^{-k} = [c(s+t)]^{-k} \qquad (2)$$

if c is defined to equal $B^{-1/k}$. Thus T may be considered a scale invariant function of t.

---

[1] Newell and Rosenbloom use trials n, not time t, as their independent variable. This is a minor complication since there is a fairly straightforward method, for adjusting k when converting from one form to the other.

The Newell and Rosenbloom measure ranges between infinity (limit of bad performance) and zero (limit of good performance) as (s+t) ranges between zero and infinity. Our OPT performance measures, on the other hand, are on a scale of 0 to 1, from bad to good. So a conversion of some sort is needed. An appealingly simple conversion would be

$$T(t) = \frac{1-P(t)}{P(t)} \qquad P(t) = \frac{1}{1 + T(t)}, \tag{3}$$

i.e., to have P in the form

$$P(t) = \frac{1}{1 + [c(s+t)]^{-k}} . \tag{4}$$

However, this is of quite a different form than the learning curve used in early OSBATS work. That form, with a scaling constant now added, is

$$P(t) = 1 - [1 + c(s+t)]^{-k} . \tag{5}$$

It turns out that if either (4) or (5) are fitted to the entry performance P_e at t=0 and criterion performance Y at $t=t_s$ that the difference between the two curves is only a few percent. Thus there is no practical reason not to use form (5), and since it is closer to the form originally embedded in developed OSBATS modules, it was chosen.

The Learning Curves in OSBATS Context

Assuming that the task in question can be trained to some level on the medium in question, we postulate a learning curve of the following form:

$$P(t) = a_r P_d(t) \tag{6}$$

where

$P(t)$ = the (entry) performance measured on the actual equipment after training an amount of time t on the training device in question. This performance approaches a_r asymptotically.

a_r = transfer index defining the fractional amount of performance, as measured in the training device environment, that transfers to performance on the actual equipment. Assumed a simple proportion.

$P_d(t)$ = the (exit) performance, measured on the training device, after training an amount of time t on the training device. By definition, this performance measure also approaches 1 asymptotically.

Consider a performance score y in the training environment and in the operational environment. We would like this to have a comparable meaning in each environment. If, for instance, that performance of a tank gunner is defined as the fraction of time he can hit a stationary target at 1000 meter range within 5 seconds of it coming into view, then it is convenient to use the same definition for a training device employing computer generated imagery. Thus a performance of 0.80 has a certain comparable meaning in each environment. It does not mean that a student who has trained to 0.80 performance on the training device can go out and score 0.80 in the actual equipment. Rather, he will be able to score $a_r(0.80)$ on the actual equipment.

Once this comparability is established, we may address instructional features and setup time reduction in terms of time shrinkage. I.e., these factors may allow an increment of performance improvement to be achieved more quickly since every

hour of training can be used more efficiently and with less wastage. More formally:

$$\text{(Time in training device)} = \frac{1}{m}\,\text{(Time in actual equipment)} \qquad (7)$$

i.e., for every hour of training required on the actual equipment to progress from performance level y_1 to performance level y_2 it requires only a fraction 1/m of that time on the training device. E.g. if it takes an hour in actual equipment training to go from a performance score of.75 to .80, it will take, say, half an hour to do the same on the training device if instructional features and reduced setup time lead to a reduction ratio of 1/2, i.e., if m=2.

If we assume that the learning curve $P_d(t)$ for the training device has the same basic form as the learning curve $P_A(t)$ for actual equipment, and that that form is scale invariant by our earlier definition, then relation (7) above may be cast as a relationship that will allow us to establish a simple correspondence between the scaling coefficients in $P_d(t)$ and $P_A(t)$. For if we construe the times mentioned in relation (7) as the times required to go from performance level y_1 to y_2, in the respective environments, and if we label these times t_1 and t_2 for the actual equipment and t_1' and t_2' for the training device, then we may rewrite (7) as:

$$\left[t_2' - t_1'\right] = \frac{1}{m}\left[t_2 - t_1\right] \qquad (8)$$

and then rewrite this as

$$\left[P_d^{-1}(y_2) - P_d^{-1}(y_1)\right] = \frac{1}{m}\left[P_A^{-1}(y_2) - P_A^{-1}(y_1)\right] \qquad (8')$$

where the -1 denotes the inverse of the functions indicated. But since we have assumed $P_A(t)$ and $P_d(t)$ have the same scale invariant form, we may write:

$$P_A(t) = f(\{c_A[s_A + t]\}, \text{ and other parameters}) \tag{9}$$

and

$$P_d(t) = f(\{c_d[s_d + t]\}, \text{ and other parameters}) \tag{10}$$

where the other parameters, including such things as the shape constant k, are common to both $P_A(t)$ and $P_d(t)$ and will be suppressed in the remaining equations. Then

$$t_2 = P_A^{-1}(y_2) = \frac{1}{c_A} f^{-1}(y_2) - s_A \tag{11}$$

and the other times in (8) and (8′) are found similarly. Substituting these into (8) yields

$$\left[\left[\frac{1}{c_d} f^{-1}(y_2)-s_d\right]-\left[\frac{1}{c_d} f^{-1}(y_1)-s_d\right]\right]$$

$$= \frac{1}{m} \left[\left[\frac{1}{c_A} f^{-1}(y_2)-s_A\right]-\left[\frac{1}{c_A} f^{-1}(y_1)-s_A\right]\right] \tag{12}$$

but this simplifies to

$$\frac{1}{c_d} = \frac{1}{m} \frac{1}{c_A}$$

or

$$c_d = m \ c_A \tag{13}$$

which is the promised correspondence between scaling constants.

The actual application of these results is as suggested in the summary. For notational simplicity we call c_A simply c. Then rather than write c_d in the equation for $P_d(t)$, we write

$$P_d(t) = f(\{mc \ [s_d + t]\}) \tag{14}$$

where s_d is selected so that $P(0) = P_e$, i.e., $P_e = a_r P_d(0)$. But substitution into (14) yields $P_d(0) = f(\{mc[s_d + 0]\}) = P_e/a_r$, from which we may conclude that

$$s_d = \frac{1}{mc} f^{-1}(P_e/a_r) \quad . \tag{15}$$

Remark: The requirement $P_d(0) = P_e/a_r$ warrants some discussion. This is a direct result of the assumption that transfer of training from the training device to actual equipment is a linear function of performance on the training device. Since the training device will certainly have less than perfect transfer (i.e., $a_r < 1$), it follows that $P_d(0) > P_e$, i.e. the entering student would be able to score better on the training device than on the actual equipment. This is not a particularly unreasonable assumption, but neither is it obviously true, especially when quantified strictly with the $1/a_r$ ratio. There may be times when this conflicts with the desired "natural" comparability interpretation between $P_A(t)$ and $P_d(t)$ discussed earlier. In that case we must either abandon the linear transfer assumption or the strict comparability interpretation. The latter compromise is preferred, since it only involves rescaling the scoring instrument for one or the other environments, rather than a major reformulation of the OPT models. A continuation of the tank gunner trainer example may prove useful. If the typical student could score 0.2 in a real tank with virtually no new training (e.g., just a few minutes spent with a manual), and if the training device is deemed to provide 0.8 transfer, then the entering student should be able to score $P_d(0) = 0.2/0.8 = 0.25$ on the training device with virtually no training (e.g., with just a few minutes with a manual). If this is not the case, it needs to be made the case, i.e., the scoring instrument is revised so that the average entering student <u>does</u> score 0.25.

We propose that the time-multiplier m be partially based on considerations of the effective use of time in the actual equipment versus the training device environments, and partially on the benefits of instructional features. Suppose training a student on take-off and landing tasks entail only 20 useful minutes out of every hour training session, the remaining time being spent on transportation to the airfield, pre-flight checkout, and holding pattern cycling--activities that constitute overtraining of mastered tasks, or just wasted time. Let u_A =

20/60 represent the ratio of useful to total time on actual equipment. Suppose, further, that 50 minutes of every hour are deemed useful on the training device in question. Let u_d=50/60 represent the the ratio of useful to total time on the training device. Then it seems the training device has an $m = u_d/u_A$ = 50/20 advantage over the actual equipment, in terms of the usefulness made of time.

We argue that instructional features provide a similar time-multiplier effect by enhancing every hour in the training device in terms of concepts presented, effective trials executed, and reinforcement and feedback provided. Let I_d be an index of any such instructional features benefit in the training device. (E.g., I_d=1.0 represents no enhancement, and I_d=1.5 represents a 50% better use of time; we expect modest enhancements for most training devices, e.g., I_d = 1.2.) Then in addition to the time-multiplier effects occasioned by reduced time wastage, the instructional features are included in m:

$$m = I_d \frac{u_d}{u_A} . \tag{20}$$

As an illustration of the interpretation of m, consider the training of takeoff and landing on a recreational flight simulator for a personal computer versus that in an actual Piper Cub aircraft. To learn this on an actual Piper Cub[2] we would need to go to the airport, hire an instructor, wait for the aircraft to be readied, be given a briefing on aircraft instrumentation, told the basics of flight, wait for runway clearance, circle around between takeoff and landing, etc.

---

[2]We are simply envisioning the rudimentary acquisition of the skill -- not a certification program requiring the mastery of navigation, regulation, flight physics, as well as the actual development of the take-off and landing skills. At the level of skill we are envisioning we would be uncertified and unwise to attempt takeoff or landing without an instructor at dual controls.

Takeoff and landing is not too hard on an actual Piper Cub (we are told) so suppose that in a 6 hour day at the airport we are able to acquire the rudimentary skill, e.g., to go from a completely green performance score of, say, 0.05, to score of 0.5 on an appropriate scoring instrument. Suppose that during that 6 hours we spend 2 hours actually being instructed on takeoff and landing, observing the instructor perform takeoff and landings, attempting the task ourself at our dual controls, and receiving feedback and critique. Then $u_A = 2/6$. In contrast, in learning to take off and land in a simulated Piper Cub on our PC-based recreational flight simulator almost all our time is devoted directly to learning the task by reading the manual or by actually attempting take offs and landings. Thus a useful time ratio $u_d=50/60$ might be reasonable for this environment. In addition, the simulated environment offers some significant instructional feature advantages over the actual environment. We can intentionally make mistakes in order to get a feel for aircraft response, stall speed, rates of descent, and so forth. Crashing the plane offers reinforcement not desirable with actual equipment. The value of such instructional features is difficult to estimate, but let us say $I_d=1.20$. Then the time-multiplier for the PC-based simulator is $m = I_d(u_d/u_A) = 1.2[(50/60)/(2/6)] = 3.0$. Since it takes 6 hours to acquire rudimentary take off and landing skill on actual equipment, it will only take 6/3 = 2 hours to acquire the rudimentary skill on the simulator (i.e., to go from entry level performance P_e/a_r to a score 0.5 on a comparable scoring instrument) under our assumptions. We should expect the transfer ratio from our PC-based flight simulator to the actual aircraft to be very poor, since the fidelity is extremely poor in every dimension; perhaps $a_r=0.20$ would be a reasonable estimate. (Estimation of transfer of training is the subject of the next section.) So we would only be able to score (0.2)(0.5) = 0.1 on an actual aircraft after spending 2 hours on the simulator. Further, we could never attain the 0.5 score achieved in 6 hours on actual aircraft by training instead only on the simulator. A perfect 1.0 rating on the simulator only translates to a 0.2 rating on the actual aircraft.

Transfer of Training

Our approach to determining the transfer of training for a medium involves two steps. First, we determine the transfer index a_R for the task in question for a medium that serves as a benchmark in the sense that it meets all the fidelity requirements exactly. Second, we degrade that transfer index to attain the actual transfer for the medium under consideration, which may be in some ways better and in other ways worse than the benchmark medium. Both of these steps are based on measures of fidelity relative to the actual equipment and operational environment. The fidelity dimensions are indexed by i. The benchmark medium has fidelities labeled $R_1, \ldots, R_n$. The medium in question has fidelities labeled $r_1, \ldots, r_n$.

Benchmark Fidelities. An essential part of the OSBATS data base is the definition of the benchmark medium in terms of fidelity levels. In work to date we have called these the fidelity requirements for the task. Originally we defined fidelity requirements as follows:

> Definition 1: Required fidelity represents the minimum level of realism (in the various fidelity dimensions) required to train the task to standard on the training device.

Mathematically, this was interpreted to mean that the learning curve P(t) had asymptote at the standard Y:

$$\lim_{t \to \text{Infinity}} P(t) = Y \quad . \tag{21}$$

But, as is clear from (6), P(t) also approaches a_R, the transfer index from the training device to the actual equipment, so this definition was equivalent to saying that the transfer index is equal to the training standard if all fidelity requirements are just met:

$$a_R = Y. \tag{22}$$

This definition was fairly simple and was dependent only on the nature of the task, not on any assumed features of the training device (instructional features, reduced setup time, etc.). However, it seemed to us that the statement "fidelity requirements met" should imply that the device should be able to train to the standard in a finite amount of time, rather than simply approaching it asymptotically. Therefore a second definition was considered:

Definition 2. Required fidelity represents the level of realism needed to train the task to standard on the training device in the same time as on actual equipment, assuming that the training device eliminates most (nominally 60%) of the setup time and will have instructional features causing further reduction (nominally 10%) in the time required to train tasks on the device as compared to on actual equipment. Thus, if fidelity requirements are met, the increased time to train to standard due to lower fidelity is just offset by instructional features and device time savings due to reduced setup time.

Mathematically this was equivalent to the requirement that

$$P(t_S) = Y \tag{23}$$

where t_S is the time required (including setup) to train the task on actual equipment. This definition had the advantage of allowing standard achievement in a finite amount of time on a training device meeting the fidelity requirements. However, it had five major disadvantages: (1) it was difficult to understand and to communicate to a SME; (2) it required consideration and knowledge of setup time on the actual equipment; (3) it required an assumption on the nature of the training device, in terms of setup time savings and instructional features benefits; (4) it

became more difficult and time consuming to solve for the learning curve parameters; (5) on a task for which negligible setup time is required on actual equipment, and for a device without instructional features, essentially perfect fidelity would be required, which seems unreasonable. Therefore, this definition was ruled out.

So we reconsidered definition 1. Perhaps it would be workable if we were just careful with the associated words defining the meanings of the fidelity "requirements". Since a training device just meeting the fidelity "requirements" takes forever to reach the training standard, the medium selection methodology will end up selecting last stage media (prior to actual equipment) with some fidelity requirements exceeding these "requirements". Is this really objectionable? It just defines a floor. Media with lesser fidelity might still be selected for use in earlier training stages, or even for the final stage if other fidelity dimensions compensated. So it was suggested that we go back to definition 1, but talk about fidelity "floor", or "base", or some other more accurate term.

But the counter argument held the day: that a SME would be uncomfortable estimating such "floor" requirements--that he or she would be better able to estimate fidelities actually needed to train the task to criterion on the training device. So we proposed a variation on definition 1, namely by choosing a transfer index equal to some level of performance higher than the standard, and defining fidelity requirements so that the transfer equals that level of performance. In particular, we proposed:

<u>Definition 3: Required fidelity is achieved by a fidelity vector (R_1, ..., R_n) such that the associated transfer index a_R exceeds the training standard Y by a fraction x of the distance to the standard, i.e.,</u>

$$a_R = Y + x(1 - Y). \qquad (24)$$

This definition allows the actual standard to be achieved in a finite amount of time on a device just meeting the requirements. It introduces another parameter, x, but this extra parameter is advantageous since it allows us to adjust to the SME interpretation of the $\{R_i\}$ rather than requiring us to come up with wording and instructions to force the SME to our interpretation of the $\{R_i\}$. We just try different values of x to come up with something that corresponds to SME interpretation of the term "fidelity requirements" (or whatever other term we choose to indicate the $\{R_i\}$). Should x turn out to be 0, for the terminology used, then Definition 3 corresponds exactly to Definition 1.

Other approaches to defining the $\{R_i\}$ can be suggested[3], but for now we will go with Definition 3. The advantages are, to recapitulate: (1) allows the SME to interpret our "fidelity requirement" terminology as he or she chooses (we will adjust x appropriately); (2) depends on the task to be trained only--no assumptions are made regarding features of the actual training devices; (3) decouples training transfer from learning curve parameters, consistent with the plan outlined at the beginning of this paper, thereby eliminating the complex and time consuming iteration occasioned in some of our earlier formulations.

Transfer Index Degradation. Let R_i be the "required" (or "benchmark") fidelity (i-th dimension) for a training device under consideration to train the task in question, according to SME judgment. Let r_i be the actual fidelity (i-th dimension) for the training device in question. Assume, for now, that we are looking for a function

---

[3]One approach suggested defining the $\{R_i\}$ such that if the corresponding fidelities in the actual equipment were degraded to those levels (down from 1.0) then the task training would take twice (or some other set multiple) as long.

Yet another approach suggested assuming that the SME were generous in estimating each "required fidelity", so we would degrade the SME estimate by some fixed percentage to obtain the $\{R_i\}$ to be used with Definition 1.

$a_r = f(r_1, \ldots, r_n; a_R, R_1, \ldots, R_n)$ that satisfies the following conditions:

(1) $a_r = 0$ if any $r_i = 0$ when the corresponding $R_i > 0$, i.e., if a fidelity dimension is included at all, then it is assumed that some fidelity in this dimension is absolutely essential. (This condition is arguable, and a variation on it will be considered later.)

(2) $a_r = a_R$ if each $r_i = R_i$. (I.e., any device with the same fidelity measures is indistinguishable from the benchmark device defined by the SME, insofar as transfer of training is concerned.)

(3) $a_r = 1$ if each $r_i = 1$, i.e. perfect fidelity is achieved. (In this case the device is indistinguishable from the actual equipment, insofar as transfer of training is concerned.)

Simple Multiplicative Axis-Anchored Degradation Function. A very simple function that satisfies the requirements is

$$a_r = \left[\prod_i r_i\right]^Q, \qquad \text{where } Q = \frac{\ln a_R}{\prod_i R_i}. \tag{25}$$

where the "pi" indicates a mathematical product of the terms following.

Relaxed Axis-Anchored Multiplicative Degradation Function. A related functional form uses the required (benchmark) fidelity dimensions as exponents to give a certain "softening" in response to degradation in less demanding requirements. The functional form is

$$a_r = \left[\prod_i r_i^{R_i}\right]^Q, \qquad \text{where } Q = \frac{\ln a_R}{\prod_i R_i^{R_i}}. \tag{26}$$

The softening is shown by differentiating both (25) and (26) with respect to r_i. In the first case,

$$\frac{da_r}{dr_i} = Q \frac{a_r}{r_i} ,$$

while in the second case,

$$\frac{da_r}{dr_i} = Q R_i \frac{a_r}{r_i} .$$

These equations are more useful for this discussion when rewritten in terms of the fractional change (dr_i/r_i) in r_i, and the associated fractional change (da_r/a_r) in a_r, i.e., the ratios of the differential elements to their respective variables. For the simple multiplicative axis-anchored degradation function, the relationship is

$$\frac{da_r}{a_r} = Q \frac{dr_i}{r_i} , \tag{27a}$$

while for the relaxed axis-anchored multiplicative degradation function the relationship is

$$\frac{da_r}{a_r} = Q R_i \frac{dr_i}{r_i} . \tag{27b}$$

Thus, a 5% change in r_i will effect the same percentage change (5% of Q) in a_r regardless of the value of R_i, for the simple multiplicative axis-anchored degradation function, but will effect a percentage change (5% of Q R_i) in a_R proportional to R_i for the relaxed axis-anchored multiplicative degradation function. Thus, a_r will be less sensitive to percentage shortfalls in R_i when R_i is low, in the latter case.

<u>Un-Anchoring at the Axes</u>. Suppose that condition (1) is replaced by conditions on each fidelity dimension indicating whether, and to what degree, degradation is limited as that

fidelity is driven to zero. Let us formalize this by the following rule: Hold $r_j = R_j$ for each fidelity dimension j except one, i. Consider the response in a_r to variations in r_i. As r_i varies from 1.0, down through R_i, and on down to 0, the value of a_r varies from some maximum value, a_{ri}^+, to a_R, to a minimum value, ar_i^-. Require that a_{ri}^-/a_{ri}^+ equal a fraction f_i, characteristic of that fidelity dimension and the task, or general type of task to be trained. Setting each $f_i = 0$ is equivalent to restoring condition (1).

The new rule allows certain fidelity dimensions to be deemed less critical than others. For instance, it seems that for most tasks audio fidelity is more of an enhancement than an essential. It might therefore be reasonable to set $f_i = 0.3$, say. Although in principle there will be a distinct set of $\{f_i\}$ for each task, it is hoped that the f_i will be not vary over large groups of tasks, so that SME estimation workload will not be substantially increased. This hope is bolstered by the following interpretation of the $\{f_i\}$, which can be shown to be true: $f_i = a_r$ if $r_i = 0$, and $r_j = 1$ for $j \neq i$. But since a device with $r_j = 1$ for all j is indistinguishable from actual equipment in regards to training, this suggests a way to more concretely think about f_i. Suppose a training device is constructed by taking actual equipment, but somehow denying the student the i-th fidelity dimension; then f_i is the transfer of training that would occur from this device to actual equipment. (In order to avoid safety implications, assume that an instructor-operator has dual controls, as needed.) For instance, if the student were giver heavy, padded headphones that prevented helicopter noise from being heard, yet still permitted communication with an instructor, how well would an actual helicopter train:

(1) Instrument navigation? (Answer: Fine. $f_i = 1$.)

(2) Take-off and landing? (Answer: Reasonably well, though some useful information contained in engine and

rotor noise and would be unavailable in this training. So perhaps $f_i = .6$.)

(3) Engine or drive train malfunction? (Answer: Very poorly. This kind of malfunction is very often determined and diagnosed by sound, so perhaps $f_i = 0$.)

We might, similarly, consider how well the student training would transfer if his training were conducted with the windows covered with gauze, giving the effect of highly impaired visual resolution, and thereby estimating the associated f_i. If we envision rigging up mechanical assists or detriments to destroy the proper control loading feel of actual equipment control actuators (stick, levers, knobs), then we may estimate f_i for the control loading fidelity dimension. Shuttering the windows immediately after weapon firing and before warhead impact would provide us a basis for estimating f_i for the weapon effect fidelity dimension. Not all fidelity dimensions can be addressed in this way; one cannot effectively degrade or deny motion cues in actual equipment, for instance.

The particular degradation function associated with this new condition is:

$$a_r = q \left[\prod_i (r_i + b_i)^{R_i} \right]^Q \tag{28}$$

where q, Q, and the $\{b_i\}$ are parameters selected so that a_r satisfies the operable conditions, viz.,

(1′) For each i, $a_{ri}^- = f_i \, a_{ri}^+$, where a_{ri}^- is the transfer associated with $r_i = 0$ and where a_{ri}^+ is the transfer associated with $r_i = 1$, when $r_j = R_j$ for each fidelity dimension j other than i.

(1") [Equivalent to (1′)]. If $r_i = 0$, and $r_j = 1$ for each j $\neq$ i, then $a_r = f_i$.

(2) $a_r = a_R$ if each $r_i = R_i$. (I.e., any device with the same fidelity measures is indistinguishable from the benchmark device defined by the SME, insofar as transfer of training is concerned.)

(3) $a_r = 1$ if each $r_i = 1$, i.e. perfect fidelity is achieved. (In this case the device is indistinguishable from the actual equipment, insofar as transfer of training is concerned.)

The parameters are found in the order Q, $\{b\}_i$, q, as follow. First, Q is found in order to force

$$a_R = \prod_i \left[R_i + (1 - R_i) f_i^{\frac{1}{R_i Q}} \right]^{R_i Q} . \qquad (29)$$

(Unfortunately, this requires an iterative root finding approach. However, it is needed only to find the parameters for the a_r curve, not for each evaluation of a_r.) Next, find each of the $\{b_i\}$, using the formula:

$$b_i = \frac{f_i^{\frac{1}{R_i Q}}}{1 - f_i^{\frac{1}{R_i Q}}} \qquad (30)$$

The fitting is completed by finding q from

$$q = \left[\prod_i (1 + b_i)^{R_i} \right]^{-Q} . \qquad (31)$$

A short exercise in algebra would show that if each f_i equals 0 that this formulation collapses to that of the relaxed axis-anchored degradation function. This is quite reasonable, since in that case condition (1′) or its equivalent (1″) degenerate to the axis-anchoring condition (1) presented earlier.

Transfer Degradation Based on Analogy to Utility Theory. We also investigated the use of degradation formulas from utility theory, as developed by Keeney and Raiffa (1976), since there, too, one is estimating a function between 0 and 1 on some space. We hoped that some of the independence conditions that prove useful in utility theory would have meaningful analogues in the fidelity degradation problem. However, we have not been successful in these attempts. It appears that there are inherent conflicts in the two formulations. Further research along these lines is being held in abeyance.

Detailed Computational Example

We conclude the discussion of measurement effectiveness modeling with a detailed example employing modified power function learning curves and using a degradation function that is not anchored at all to the axes. Suppose there are two fidelity dimensions of interest. Suppose that the following estimates are provided by the SME and/or design engineers:

Fidelity Dimension	Required (Benchmark) Fidelity Level	Minimum Performance Parameter	Estimated Training Device Fidelity Level
i	R_i	f_i	r_i
1	0.90	0.00	0.60
2	0.70	0.30	0.40

Task Training Standard:	$Y = 0.75$
Estimated Entry Level Performance:	$P_e = 0.10$
Time to Achieve Standard on Actual Equipment:	$t_s = 10.0$ hrs
Useful Time Ratio, Actual Equipment:	$U_A = 0.20$

Useful Time Ratio, Training Device:	$U_d = 0.80$
Instructional Features Factor, Actual Equipment:	$I_A = 1.00$
Instructional Features Factor, Training Device:	$I_d = 1.25$

Further, suppose that through experience the OSBATS modelers have established that the parameter value $k = 1.25$ causes the learning curves to have a "shape" that seems to describe the task training processes being investigated. Also, suppose that they have established that the parameter value $x = 0.3$ seems to give proper meaning to the SME estimates of "required" fidelity.

Following the upper path of the IDEF diagram shown at the beginning of this memorandum, we first find from (24) that $a_R = 0.825$.

Next, the parameters Q, b_1, b_2, and q for the a_r relation are found. Solving (30) with a root finding procedure yields Q = 0.5669. From (31), we find that $b_1 = 0$ and $b_2 = 0.0176$. From (31) we then find that q = 0.9931. Then the transfer function for training devices as a function of the fidelity levels r_1 and r_2 is found from (28):

$$a_r = 0.9931 \left[(r_1 + 0)^{0.9} (r_2 + 0.0176)^{0.7} \right]^{0.5669}$$

where the exponents 0.9 and 0.7 are R_1 and R_2, respectively. A graph of a_r as a function of r_1 for various levels of r_2 is presented in Figure B-2.

We have assumed that the actual equipment has a learning curve described by a modified power function (5):

$$\begin{aligned} P_A(t) &= f(\{c\,[s_A + t]\}) \\ &= 1 - [1 + c(s_A + t)]^{-k} \end{aligned}$$

where f has the functional form

$$f(z) = 1 - [1 + z]^{-k}$$

and inverse

$$z = f^{-1}(y) = (1 - y)^{-1/k} - 1 \quad .$$

We have established that $k = 1.25$, so we may determine the value of c that causes $P_A(t)$ to achieve the training standard $Y = 0.75$ after time $t_s = 10.0$ hours from entry. To do this we note that

$$c(s_A + 0) = (1 - P_e)^{-1/k} - 1$$

and

$$c(s_A + t_s) = (1 - Y)^{-1/k} - 1$$

which we may solve simultaneously to yield the basic time scaling factor:

$$c = \frac{(1 - Y)^{-1/k} - (1 - P_e)^{-1/k}}{t_s} = 0.194$$

and the head start for actual equipment:

$$s_A = \frac{1}{c} \left[(1 - P_e)^{-1/k} - 1 \right] = 0.452.$$

We assume that the "local" performance curve $P_d(t)$ for the training device has the form

$$\begin{aligned} P_d(t) &= f(\{mc\ [s_d + t]\}) \\ &= 1 - [1 + mc(s_d + t)]^{-k} \end{aligned}$$

where f() is the same as for actual equipment. We determine the the time-multiplier as:

$$m = I_d \frac{u_d}{u_A} = 5.00$$

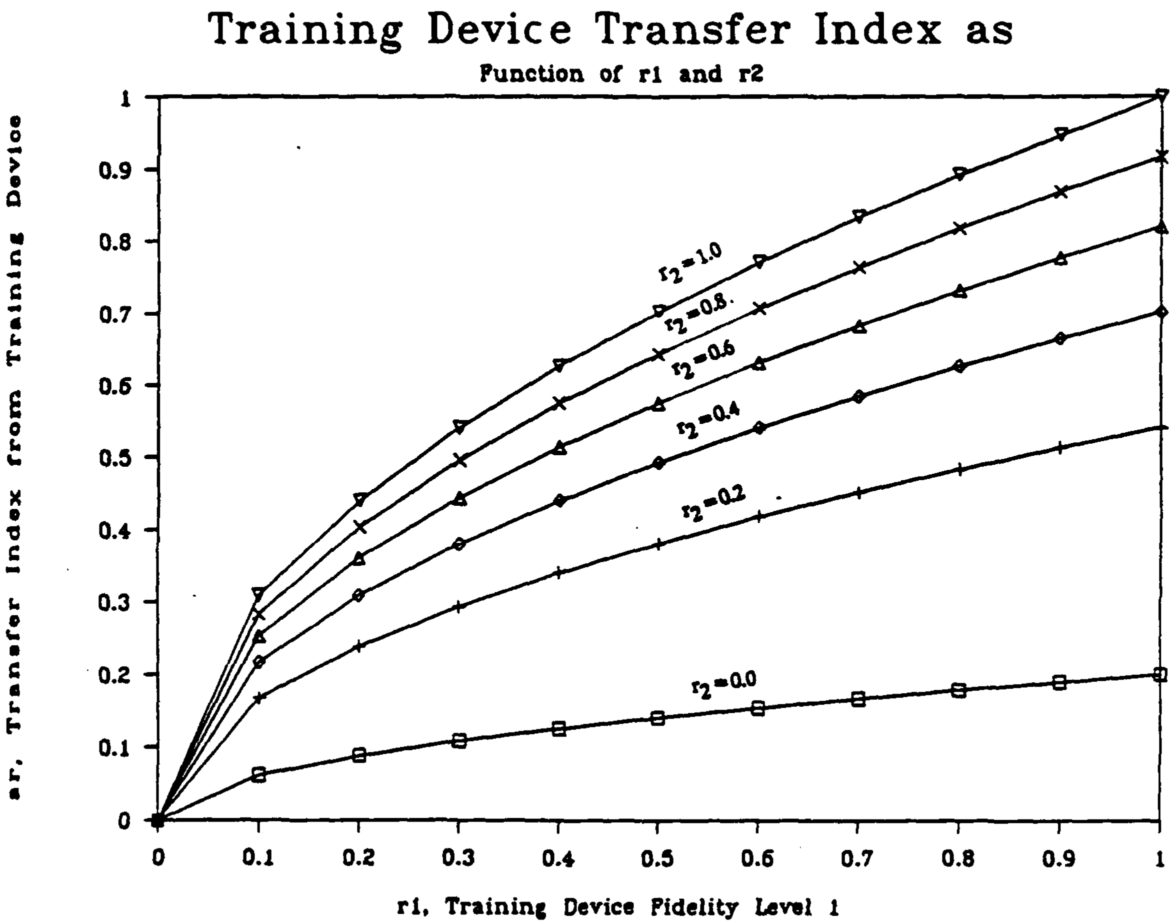


Figure B-2. Training Device Transfer Ratio as Function of Two Fidelity Levels

and the head start as:

$$s_d = \frac{1}{mc} f^{-1}(P_e/a_r) = \frac{1}{mc}\left[(1 - P_e/a_r)^{-1/k} - 1\right] = 0.183.$$

Remark: The only iterative procedure needed in this example was the determination of the parameter Q for the function a_r. As various training devices options are considered, only the $\{r_i\}$, not the $\{R_i\}$, are varied, so this iterative procedure need not be repeated. This is an improvement over some of our previous models in which speed of computation was a potentially serious problem.

The learning curves $P_A(t)$, $P_d(t)$, and $P(t)$, $P_d(t)$ are shown in Figure B-3. Note that $Pd(0) > PA(0) = P(0)$, as discussed earlier. As expected, P(t) provides the greatest training benefit initially, but then becomes relatively ineffective due to the transfer index "ceiling".

Cost Functions

Summary

Two types of training device cost function types are employed in OSBATS modeling. The Resource Allocation module employs piecewise continuous cost functions, capable of representing setup costs, the jumps occasioned by discrete acquisition of major cost training devices, and the implications of training devices already on hand. This type of function can also be used to represent economies of scale, though this has not been explicitly addressed in work to date. The detailed cost function is based on training system per unit acquisition costs, life cycle, fixed operating costs, variable operating costs, and hours that the device may be used annually. The relationship of the cost function to these constituent components will be explained. The methodology for determining the values of the

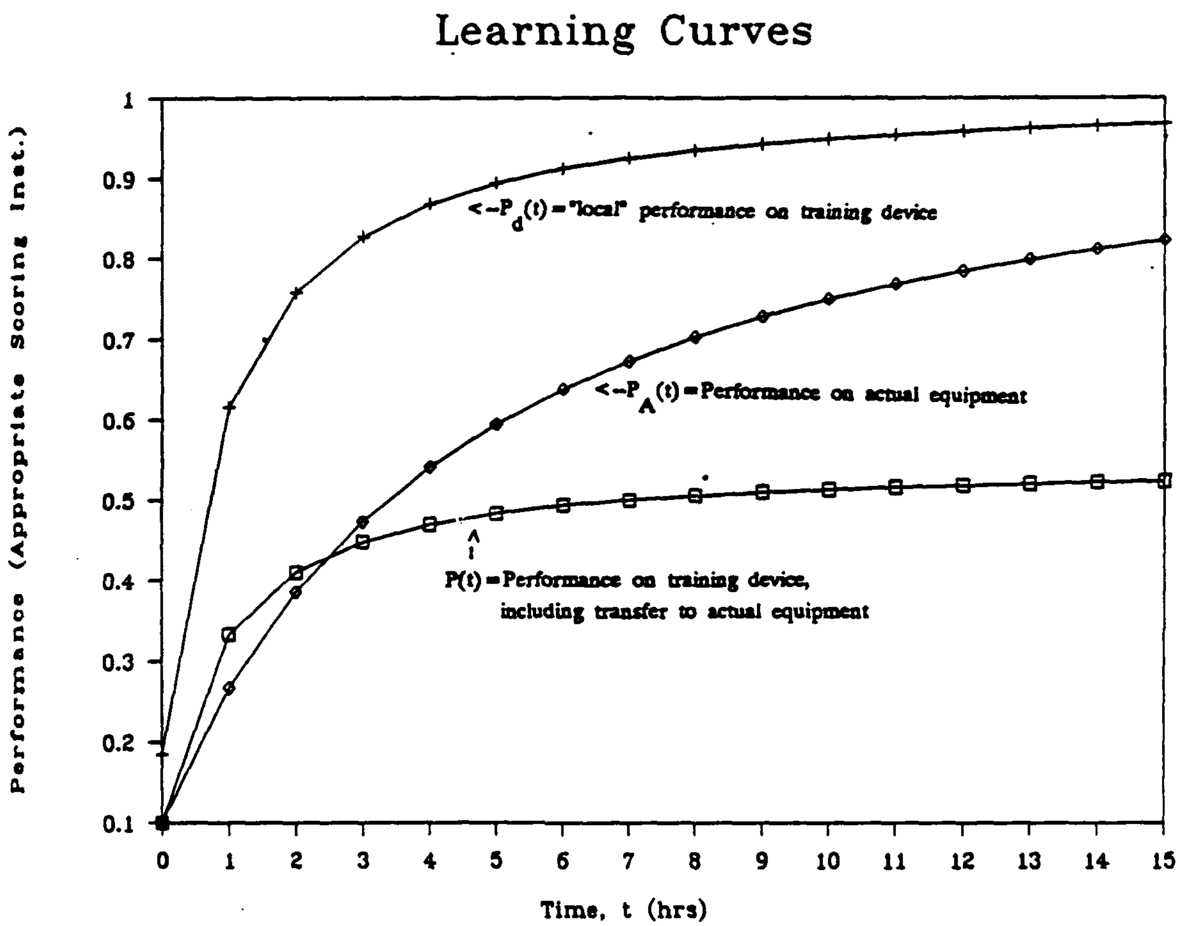


Figure B-3. Learning curves.

constituent components themselves are not addressed in this appendix.

The other modules employ a simple linear cost function based on a fixed hourly rate. This simple cost function is an approximation to the more detailed function.

Detailed Cost Function

The piecewise linear cost curve consists of a series of linear segments. In the case that economies of scale are unimportant, only the discrete acquisition of devices causes deviations from simple linear curve, and results in a cost curve with an infinite series of equal steps. The cost curve is formally defined by the following parameters:

t - Expected total time that an individual student will use the training device in question during his training program.

C(t) - Training device costs attributed to each individual student if each student uses training device for amount of time t.

N - Number of segments in the cost curve.

i - Index of segment (interval) of cost curve.

j - Total number of devices acquired (for given t).

T_i - Starting value for i-th segment of cost curve C(t)

A_i - Height of cost curve C(t) at beginning of i-th segment.

B_i - Slope of i-th segment of cost curve C(t).

With two exceptions, each segment is open on the left and closed on the right. The first exception is the first interval, which is closed on the left and on the right. The second exception is the final interval, which is open on the left and extends without limit on the right. The cost curve is defined as follows:

$$C(t) = \begin{bmatrix} A_1 + B_1(t - T_1) & \text{if } T_1 <= t <= T_2 \\ A_i + B_i(t - T_i) & \text{if } T_i < t <= T_{i+1} \\ A_N + B_N(t - T_N) & \text{if } T_N < t \end{bmatrix} \qquad (32)$$

The parameters A_i and B_i may be related to the following data or projections:

P - Number of students trained annually in the training program.

I - Investment cost for each training device.

L - Life cycle for each training device.

F - Fixed operations cost, annual.

r - Variable operations cost, per student hour.

H - Maximum potential annual usage (student hours) of device per year, considering school operating hours, device maintenance schedules, number of students served simultaneously, etc.

These parameters are related to the cost curve C(t) by the following master equation:

$$C(t) = \frac{\text{Total investment cost} + \text{Total fixed operating cost} + \text{Total variable operating costs}}{\text{Total number of students processed}}$$

where "total" means across all students and all years in the device lifetime.

The total number of students processed over the life cycle of the training device is just PL.

Suppose that each student uses the kind of training device in question for t hours. Then since P students are processed annually, the training device will be used a total of Pt hours annually. But each particular training device is available for only H hours annually, so the total number of devices required is

$$j = \left[\begin{array}{ll} Pt/H & \text{if } Pt=jH \text{ (exact multiple)} \\ \text{integer}(Pt/H)+1 & \text{otherwise} \end{array} \right] \quad (34)$$

This will remain in effect for an interval of length H/P, at which time j is incremented. So the steps will occur at $T_1 = 0$, $T_2 = Pt/H$, ..., $T_N = (N-1)Pt/H$.

If some training devices of the type in question have already been acquired, then the cost analyst will want to make sure that the steps associated with acquisition of these first devices properly reflect both sunk investment cost and sunk (obligated) fixed costs. Essentially, this will remove the steps in the cost curve that would otherwise have been associated with the acquisition of the devices now on hand. This modification will be quite important for proper operation of the Resource Allocation module, since will result in a proper tendency to allocate more heavily to these devices. Because the present OSBATS data base assumes assets on hand are zero, this refinement will not be discussed further here.

If j devices are acquired, the total investment cost is jI (since the j training devices are purchased only once during the

life cycle). The total fixed annual operating cost is jFL(since the annual fixed operating costs must be expended for each device and each of the years in the device life cycle). The variable operating cost is PLrt (since the variable operating cost rate r is imposed only for the PLt student hours expended over the life cycle of the training device).

Then the total cost is

$$C(t) = \frac{jI + jFL + PLrt}{PL} \tag{35}$$

Assume that the first segment of the cost curve is simply the single point at (0,0) corresponding to no devices being acquired. The second segment of the cost curve then corresponds to one device being acquired. In general, the i-th segment will correspond to i-1 devices being acquired, i.e.,

$$j = i-1 \tag{36}$$

But the equation for the i-th segment of the cost curve is given by (1) as $C(t) = A_i + B_i(t - T_i)$. Equating this to (3) and then equating the coefficients of t gives:

$$B_i = r \tag{37}$$

and then equating the constant terms gives:

$$A_i = (I/L + F)j/P + rT_i \, . \tag{38}$$

In this way the familiar cost components translate into a piecewise continuous cost curve of equal steps occurring at equal intervals and with the same slope for all segments--i.e., there is no economy of scale incorporated.

It is desirable to represent some number of the discrete segments of such a cost curve, but to smooth the rest of the curve into a single linear segment. The slope for this segment will be

$$B_N = \frac{A_i - A_{i-1}}{T_i - T_{i-1}} = (I/L + F) + r \tag{39}$$

The starting value for this last segment may be taken as the midpoint between where the last segment ends and where the next segment would have begun using the normal step:

$$A_N = A_{N-1} + B_{N-1}(T_N - T_{N-1}) + (I/L+F)/(2P) \ . \tag{40}$$

The decision to transition to the smooth curve approximation is based on the cost curve step height:

$$\text{Step height} = (I/L+F)/P \ . \tag{41}$$

If this height is small relative to that of other devices, or relative to the slope associated with variable operating costs, then the transition may be made after only one or two segments, or perhaps the curve can be treated as simple linear from the start. "Big ticket" items such as full mission simulators, on the other hand, occasion large steps, so five or more segments may be desirable before transitioning to the smooth approximation. In any case, the total number of combinations across all devices, defined as the product of the number of segments in the associated cost curves, must be kept to some manageable level -- less than 100, say -- or the solution time for the Resource Allocation module will become unacceptable.

Simple Linear Cost Function

The simple linear cost function employed by the other OSBATS modules is simply that obtained by applying (7) from the start, i.e., approximating the cost curve as simple linear with slope

$$B = (I/L + F) + r \quad . \tag{42}$$

Optimization Problems and Techniques

Summary

The general optimization problem addressed by OSBATS is to determine, for each task to be trained, the sequence of training devices, and the associated allocation of student time, that will cause the student to achieve criterion performance at minimum total cost. (The algorithms developed to solve this problem may also be used explore the complementary problem: to maximize training achieved for a given fixed cost.)

An implicit assumption underlying the OSBATS models and associated optimization algorithms is that when a student switches from one training device to another he or she transfers all of his or her operational performance score attained to date. Without this assumption the optimization problem becomes much more complex.

This general optimization problem may be cast as a sequence of increasingly more complex, difficult, and realistic problems, graded both by the number of tasks considered, the complexity of the cost function, and the existence of allocation or performance constraints. OSBATS employs the simplified problems and associated algorithms in the earlier design modules. The more complex problems are tackled in the Resource Allocation module. From easiest to most difficult, the hierarchy of optimization problems actually or potentially underlying OSBATS is as follows:

- Single-task problem with simple linear cost function, and with no allocation or entry level constraints.

- Single-task problem with piecewise linear cost function, and with allocation and entry level constraints.

- Single-task problem with continuously nonlinear cost function, and with allocation and entry level constraints.

- Multiple-task problem with simple linear cost function, and with no allocation or entry level constraints.

- Multiple-task problem with piecewise linear cost function, plus allocation and entry level constraints.

Three optimization approaches have been considered in tackling these problems. The first is an analytic technique based on necessary conditions found from calculus, it is very fast and generally optimal for the first class of problem, though global optimality is not guaranteed. The second technique is dynamic programming, which is optimal for any of the single-task problems, but which is much more time consuming than the analytic method. The third technique is to employ heuristic procedures, coupled with the analytic optimization technique. This approach should provide good solutions in most cases.

Although dynamic programming has been used in past model development, and in checking heuristic approaches to the more complex single-task problems, it is not employed in the delivered OSBATS model. Therefore, it will not be described at this time.

The following exposition will focus on the simplest problem and the analytic approach to its solution, and will then indicate

the modifications and heuristics employed to adapt this analytic model to the more complex problems.

Single-Task Optimization Problem with Simple Linear Cost Function

This problem, which lies at the heart of the Training Device Selection module, is to determine the sequence of training devices, and the associated allocation of student time, that will cause the student to achieve criterion performance at minimum total cost, if only one task is considered for training. No constraints (other than nonnegativity) are placed on allocations of student time or on entry performance to a task.

Algorithm Description. The general strategy of the analytic algoritnm is to move from one incumbent training device to the next, beginning with that device that gives the greatest operational[4] initial performance increase per dollar. A switch from the incumbent device to a successor is made when the successor's rate of increase per dollar catches up to, and prepares to pass, that of the current incumbent device. Each crossover point is determined by finding the performance level at which the derivative of performance with respect to per-student dollar costs for the incumbent training device is matched by that of a challenger device. At this point the challenger becomes the incumbent and the incumbent is removed from further consideration.

The algorithm may be implemented either with time running forward or time running backward. For this problem, with simple linear cost functions and no entry or resource constraints, the solution will be the same for either implementation.

---

[4]The term "operational" performance is used to indicate that the this is the performance the student would score if tested on actual equipment. It is the P(t) defined at the beginning of this appendix, and includes the transfer ratio a_r, also defined earlier.

Forward Time Implementation. If time is to run forward, the specific steps are detailed as follow:

Step 0: Let $y_{now} = P_e$, i.e., set the current performance level equal to expected student entry level for the task.

Step 1: Choose as the initial training device that device i having the greatest initial rate of performance increase with respect to per-student dollar expenditures, D. That is, choose the device i for which

$$\left.\frac{dy}{dD}\right|_{i,y} = \frac{a_i\ k\ m_i \cdot c}{h_i}\ (1 - y/a_i)^{\frac{k+1}{k}}$$

is greatest. The selected device becomes the incumbent.

Step 2: For each "surviving" training device j, determine the crossover performance level y_j at which $\left.\frac{dy}{dD}\right|_{i,y} = \left.\frac{dy}{dD}\right|_{j,y}$

i.e., at which the performance improvement rate for device j matches that of the incumbent. This crossover performance level is calculated from the following formula[5]:

$$y_j = \frac{a_j - a_i z}{1 - z}$$

where

$$z = \frac{a_j}{a_i}\left[\frac{a_i}{a_j}\ \frac{m_i}{m_j}\ \frac{h_j}{h_i}\right]^{\frac{r}{r+1}}$$

[5]An equivalent formula is

$$y_j = a_i a_j\ (1 - z)/(a_j - a_i z)$$

where

$$z = [(a_j/a_i)(m_j/m_i)(h_i/h_j)]^{r/(r+1)} \quad .$$

Step 2a: If y_j does not exist, or $y_j < y_{now}$, remove training device j from further consideration.

Step 2b: If no candidate training device survives, go to Step 4 to terminate.

Step 3: Let the next incumbent be that training device j for which $y_{now} < y_j < Y$ (the task training standard) and for which $y_j \le y_k$ for every other surviving training device k. Determine time t_i required on the incumbent training device as

$$t_i = \frac{1}{m_i\ c}\left[(1 - y_j/a_i)^{-1/r} - (1 - y_{now}/a_i)^{-1/r}\right]$$

The cost of training on the incumbent device is just $h_i t_i$. Let j become the new incumbent training device and set the current performance level at the crossover performance, i.e., set $i = j$ and $y_{now} = y_j$.

Then go to Step 2.

Step 3a: If no such training device j exists, then go to Step 4 to terminate.

Step 4: Train on the incumbent training device until the training standard Y is achieved. Allocate training time

$$t_i = \frac{1}{m_i\ c}\left[(1 - Y/a_i)^{-1/r} - (1 - y_{now}/a_i)^{-1/r}\right]$$

for this purpose. The per-student cost of training on this device is $h_i t_i$.

Backward Time Implementation. The problem may be solved in backward time, deallocating training resources until performance is brought from the training standard down to the entry performance level. For this essentially unconstrained optimization problem the solution will be identical to that obtained in the forward time implementation. Such a backward time implementation may seem unnatural, but is a device used in

dynamic programming, a mathematical optimization technique. It is found useful in OSBATS when solving resource allocation problems with allocation and performance entry constraints, which will be discussed later.

<u>Algorithm Derivation and Justification</u>. The various formulas used in the algorithm will now be justified. To begin, we restate the basic form of the i-th learning curve:

$$y = P_i(t) = a_i \left[1 - [1 + m_i c\ (S_i + t)]^{-r} \right]$$

where S_i is the head start used to adjust $P_i(0)$ to a desired beginning, or reference, performance level. The inverse learning curve is given by

$$t = P_i^{-1}(y) = \frac{1}{m_i c} \left[(1 - y/a_i)^{-1/r} - 1_i \right] - S_i \ .$$

If device i is used to train between two performance levels, then the time to go from the one performance level, y_1, to another performance level, y_2, is found as the difference $t_i = t_2 - t_1$:

$$t_i = P_i^{-1}(y_2) - P_i^{-1}(y_1) = \frac{1}{m_i c} \left[(1 - y_2/a_i)^{-1/r} - (1 - y_1/a_i)^{-1/r} \right].$$

This formula will be seen twice in the algorithm formulas, above. This is a particularly convenient formula in that the head start S_i has cancelled out. By focusing the algorithm on performance levels rather than times, this complicating variable is avoided.

The transition from one device to another occurs when the derivatives of performance with respect to dollars are equal. The reasoning for this is simple. We want to increase performance as much as possible for any expenditure of student time (and the associated costs). Since we may switch from one training device to another at any time without penalty, we will simply want to "ride" that device giving the highest rate of

improvement. Since performance is a continuous function of per-student attributed costs for all devices, the rate of improvement on one device can only overtake the rate of improvement on another device by passing through a performance level at which the rates of improvement on the two devices are equal. To determine this crossover performance level, equate the expressions for the rates of improvement on the two devices, and solve for the performance level. First, we need to express performance on device i as a function of per-student costs D_i. Since $D_i = t_i h_i$, where t_i is per-student time spent on device i, we substitute $t_i = D_i/h_i$ into the learning curve formula to give, after some simple algebraic manipulation:

$$y\ (D_i) = a_i \left[1 - [1 + \frac{m_i c}{h_i} (S_i + D_i)]^{-r} \right] .$$

(It is useful to note, at this point, that this may be solved for the second bracketed term to give

$$[1 + \frac{m_i c}{h_i} (S_i + D_i)] = (1 - y/a_i)^{-1/r}$$

a relation that will prove useful shortly.)

Now differentiate $P_i(D_i)$ with respect to D_i to give

$$\frac{dy}{dD_i} = a_i r \frac{m_i c}{h_i} [1 + \frac{m_i c}{h_i} (S_i + D_i)]^{-r-1} .$$

Combining the last two equations permits us to simplify to

$$\frac{dy}{dD_i} = a_i r \frac{m_i c}{h_i} [1 - y/a_i]^{1+1/r} .$$

A similar expression holds for another device j. The crossover performance between the two devices is found by setting

$$\frac{dy}{dD_i} = \frac{dy}{dD_j}$$

or

$$a_i r \frac{m_i c}{h_i} [1 - y/a_i]^{1+1/r} = a_j r \frac{m_j c}{h_j} [1 - y/a_j]^{1+1/r}$$

which may be solved for y to yield

$$y_j = a_i a_j \frac{a_j - a_i z}{1 - z}$$

where

$$z = \frac{a_j}{a_i} \left[\frac{a_i}{a_j} \frac{m_i}{m_j} \frac{h_j}{h_i} \right]^{\frac{r}{r+1}}$$

the expression given in the algorithm.

Special Cases. Note that the crossover solution, given by the above equations, is indeterminate if $z = 1$. This will occur if device i and j have identical cost rates and learning curves. It will also occur if $m_i/h_i = m_j/h_j$ and $a_i = a_j$; in this case the lesser training rate of the one device is just compensated by its lesser cost rate. It is conceivable, though unlikely, that two devices that are indistinguishable, in this sense, might be generated in OSBATS. The current version of OSBATS makes no provision to properly handle the transition between such devices. Since the optimization logic to handle this situation would be rather complex, it is suggested that a simpler expedient be employed in future OSBATS work, should this be deemed a problem. The idea is to (1) test device parameters to determine if any two devices are identical, and, if so, to (2) randomly perturb the parameters for one device by a small, inconsequential amount, say by multiplicative factors drawn from a normal distribution with mean 1 and variance 0.00001. This tie-breaking procedure would be sufficient to allow use of the optimization algorithms described here without more complex logic.

Another situation of theoretical concern is that of multiple points of identical performance per dollar for two devices. That

cannot occur with the restricted family of learning curves employed in OSBATS.

Single-Task Optimization Problem with Piecewise Linear Cost Functions, plus Allocation and Device Entry Performance Constraints

This problem may be stated as follows: Find the training times $\{t_i\}$ for a prescribed sequence of training devices $\{i\}$ so that the student is trained from the task entry performance level P_e to the standard Y at minimum total training cost C, subject to minimum entry performance constraints $\{y_i \geq e_{i+1}\}$, and maximum usage constraints $\{t_i \leq U_i\}$.

This problem has been solved using dynamic programming, but solution time was such as to preclude interactive employment in OSBATS. Therefore, an analytic-based heuristic algorithm has been developed. This algorithm looks at every possible combination of cost curve segments across the different devices and solves the optimization subproblem with training time allocations to the various devices restricted to the respective cost segments. The first combination considered will have usage of each device restricted to a particular segment of the respective cost curve. The second combination will have device usage restricted to other segments. Every possible combination is considered in this way, in principle. However, a branch-and-bound technique is used to preclude exhaustive examination of every combination, in some cases. The technique associates with each combination a minimum possible cost, namely the total cost if each device is used only as much as the value of the lower ordinate for the associated cost segment for that combination. The actual feasible cost to train for this combination will be at least as great as this lower bound value. Thus if at any point a solution has already been found whose cost is less than the lower bound cost for a given combination, that combination can be ruled out of consideration. By ordering the combinations according to their lower bound costs, it is possible to rule out all remaining

cost curve combinations, once one is found whose lower bound cost exceeds the incumbent solution cost. This technique is employed in OSBATS. However, most combinations usually must be examined in detail, so the number of segments used in defining each device cost curve must be kept small, else the total number of combinations will become too great to be considered in reasonable time.

Strictly defined, each subproblem would have the added constraints that $\{T^-_{ik} \leq t_i \leq T^+_{ik}\}$ where T^-_{ik} and T^+_{ik} are the lower and upper bounds on student time for combination k and device i. However, we may ignore the lower bound if the following assumptions are valid:

Assumptions: (1) No cost curve has a negative step between a cost curve segment and the succeeding segment. (2) No cost curve has a segment whose slope is greater than a preceding slope[6].

For suppose these assumptions are true, and suppose the subproblem algorithm finds a solution violating the lower bound for some cost curve segment under a given combination. Then the cost attributed to that device will exceed the true cost, since (1) the backward projection of the current segment necessarily lies above the preceding segments; therefore, some other combination, for which the appropriate cost curve segment is prescribed, will yield the same training effectiveness at lower cost. Since all combinations are considered, the lowest cost will be attributed to some combination, and for that combination

---

[6]The first of these assumptions is quite reasonable in any serious costing situation, since the converse, e.g., "5 apples for one dollar or 6 apples for 80 cents" would seem to have no rationale other than a bizarre promotional gimmick. The second assumption is tantamount to assuming that there are no diseconomies of scale, e.g., that devices are not being drawn from a limited pool (such as real estate land) causing the price to rise with demand. We expect that economies of scale -- rather than diseconomies -- to be operative in training device costing, so believe the assumption valid.

the lower bounds will be respected. Therefore, we may ignore the lower bound and simply impose a single upper bound constraint:

$$t_i \leq u_i \equiv \text{minimum } (T_{ik}^{+}, U_i) .$$

Suppose a particular cost curve combination has been selected. We transform the associated subproblem to one involving a simple linear cost curve <u>without setup cost</u> by setting aside the setup cost for each device, solving the subproblem using the simple linear cost rate appropriate to the assumed cost curve segment for each device, and then adding in the setup cost after the solution to the simplified subproblem has been found.

The analytic algorithm developed for solving single task problems with simple linear cost functions, discussed earlier, cannot be used, without modification, to solve the current constrained problem. The problem is that the "natural" points of economic transition from one device to another may correspond to violation of resource constraints or entry performance constraints. Thus, heuristic rules have been incorporated to prevent such violations. These rules also have been formulated to enforce the following:

<u>Assumption</u>: No training device should be used more than once in the training of a particular task. This will be true automatically for the unconstrained problem, given the assumed family of learning and cost curves.

The revised algorithm incorporates the following rules: (1) choose the best allowable device for training the task at the current performance level (where "allowable" indicates that neither performance entry nor resource usage constraints are violated), (2) if possible, use the "incumbent" device until a "natural" transition to another device is reached, based on training benefit/cost consideration (i.e., the crossover performance levels discussed earlier), but (3) if that "natural"

performance transition cannot be reached without violating constraints, then allocate only sufficient resources to the "incumbent" to bring the performance to that constraint boundary, then switch to the most efficient allowed device at that performance level.

Following these rules may result in a device being used more than once in the training sequence. To prevent this undesirable situation from persisting, the total amount of resources allocated to the device is remembered and then, on a new, fresh allocation pass, this amount of resources is forced to be used all at the same time.

One further modification is to use the backward-in-time allocation approach mentioned earlier. This was found to prevent the allocation of large amounts of student time to "entry level" devices in an attempt to make up for resource shortfalls for some other devices.

This completes the basic ideas behind the revised single-task algorithm. The details of the algorithm, presented in the body of the report, should thus be illuminated.

Single-Task Problem with Continuously Nonlinear Cost Function, and with Allocation and Entry Level Constraints.

This problem is mentioned just for the record. It can be solved using dynamic programming if the training sequence is specified. This was done in earlier OSBATS research. It was abandoned for several reasons: (1) run times promised to be much longer than with approaches based on the analytic determination of crossover performance levels; (2) it was preferrable to not pre-specify the training sequence; and (3) it was decided that the generality provided by piecewise linearity was adequate for cost curve representation.

Multiple-Task Problem with Simple Linear Cost Function, and with No Allocation or Entry Level Constraints

This problem, too, is included simply for the record. It may be solved by simply solving the constituent single-task problems and aggregating the allocations.

Multiple-Task Problem with Piecewise Linear Cost Function, Plus Allocation and Entry Level Constraints

This is the problem actually addressed by OSBATS. The algorithmic procedure is described in detail in the body of the report, and will not be repeated here. The algorithm will be briefly discussed and related to the single-task, constrained problem discussed in some depth above, and upon which it is built. As with the single-task algorithm, the multiple-task algorithm examines each of the mathematically possible combinations of cost curve segments to see if it is possible to satisfy the training requirements under that cost structure, and to determine if the total training cost is better than offered by other combinations.

The multiple-task problem is complicated by the fact that device usage constraints (user-defined or cost-segment bounds) run across all tasks. Thus, though the single-task algorithm is capable of solving a single-task problem while respecting device usage constraints, the aggregate of the single-task solutions across tasks may very likely violate these constraints. We have not been able to discover an optimal algorithm to solve this problem -- with or without dynamic programming. So we have worked to develop satisfactory heuristics.

The idea lying behind the current heuristic algorithm is that if a particular device proves so attractive, from a training-per-cost standpoint, that unconstrained optimization calls for it to be used more than allowed by the constraints,

then the device should be made less attractive by raising its price, and the unconstrained problem solved again. By adjusting device cost rates using surcharge variables, an iterative procedure should eventually bring the problem to solution, if it is feasible. If the problem is infeasible, the surcharge variables will tend to flip-flop, and no convergence will be obtained. The idea of surcharges comes from a class of nonlinear programming approaches falling under the category of "penalty" methods, or "SUMT (sequential unconstrained minimization technique)".

In setting surcharges, it is not sufficient simply to drive the usage of all devices below the constraint limits. Upon algorithm termination, surcharges greater than 1 should correspond to device usages driven down to, but not below, allowed usage. No surcharges should be (significantly) less than 1 at termination. (These conditions on the surcharges are reminiscent of the Kuhn-Tucker conditions of constrained optimization programming.) Thus, the algorithm is designed to adjust surcharges until these conditions are true, within certain tolerances.

GLOSSARY

Variables and Parameters Used in Media Effectiveness Discussion[7]

a_r = Transfer ratio for transforming performance on a training device with fidelity vector r to performance on actual equipment. [ASM_{Ti}]

a_R = Benchmark asymptote for task T on hypothetical device R. [ASM_{TR}].

b_i = Parameter, associated with fidelity dimension i, used in fitting transfer ratio equation. [b_{Tj}]

$c = c_A$ = Learning curve time scaling factor for actual equipment. [TSF_T]

c_d = Learning curve time coefficient for training device. [= TM_{Ti} TSF_T]

f( ) = General notation for mathematical function. [n.a.]

f_i = Fractional degradation in transfer that would occur as the fidelity on the i-th fidelity dimension is decreased from 1 to 0, with all other fidelity levels held at 1. [$FMIN_i$]

i = Running index identifying fidelity dimension. [j]

I_d = Index of training device instructional features benefit. [$IADJ_{Ti}$]

k = Learning curve shape parameter. [r]

m = Time multiplier used to account for increased training efficiency due to instructional features and reduced setup time. [TM_{Ti}]

P(t) = Performance on actual equipment for the task under consideration after training on the device under consideration for an amount of time t. [$P_{Ti}(t)$ when i corresponds to actual equipment]

$P_d(t)$ = Performance score on the training device in question after training a time t on the device; this is scored on a 0-1 scale and represents the score on the training device itself, and does not include any estimate of transfer to actual equipment. [= $P(t)/ASM_{Ti}$]

---

[7] Long-form names, as used in the body of this report, are identified in brackets at the end of each entry, when applicable.

P_e = Student entry performance for the task in question. [PENTRY, ENT_T]

q = Parameter (multiplicative coefficient) used to fit transfer ratio formula. [q_T]

Q = Parameter (exponent) used to fit transfer ratio formula. [Q_T]

r_i = Fidelity level characteristic of device in question on i-th fidelity dimension, as measured on 0-1 scale. [$FCAP_{ij}$]

R_i = Target, or required, fidelity level on i-th fidelity dimension in order to train task in question. [$FRQT_{Tj}$]

s = Effective head start, chosen so that P(0) or $P_d(0)$ equals the entry performance for the device in question. [HS_{Ti}]

s_A = Head start for actual equipment. [HS_{Ti} when i corresponds to actual equipment]

s_d = Head start for training device. [HS_{Ti} when i corresponds to device other than actual equipment]

t = Time allocated for each student on the device in question for training the task in question. [t]

t_s = Time required to achieve task performance standard by training on actual equipment only. [$TTOT_T$]

u_A = Ratio of useful to total training time on actual equipment. [$U_{T,AE}$]

u_d = Ratio of useful to total time on training device. [U_{Ti}]

x = Parameter used in defining meaning of $\{R_i\}$ in terms of Y. [SADJ]

y = Performance level, used as a variable. [n.a.]

Y = Criterion, or standard, performance level for task in question. [STD_T]

z = Utility variable. Used as dummy variable in defining specific functional form f(z). Also used as intermediate variable in equation for Q. [z]

Variables and Parameters Used in Cost Function Discussion

A_i - Height of cost curve C(t) at beginning of i-th segment. [A_{in}]

B_i - Slope of i-th segment of cost curve C(t). [B_{in}]

C(t) = Training device costs attributed to each individual student if each student uses training device for amount of time t. [$CUMC_i(t)$]

F - Fixed operations cost, annual. [FOC_i]

H - Maximum potential annual usage (student hours) of device per yea: , considering school operating hours, device maintenance schedules, number of students served simultaneously, etc. [$UMAX_i$]

i - Index of segment (interval) of cost curve. [n]

I - Investment cost for each training device. [INV_i]

j - Total number of devices acquired (for given t). [n]

L - Life cycle for each training device. [LC_i]

N - Number of segments in the cost curve. [N]

P - Number of students trained annually in the training program. [GRADS]

r - Variable operations cost, per student hour. [$VHRC_i$]

t = Expected total time that an individual student will use the training device in question during his training program. [t]

T_i - Starting value for i-th segment of cost curve C(t). [T_{in}]

<u>Variables Used in Discussion Optimization Problems and Techniques</u>

a_i = Asymptote for training device i. (Same as a_r, earlier.) [ASM_{Ti}]

c = Time scaling factor in learning curves. [TSF_T]

D = "Dollars", cumulative per student training costs. [n.a.]

e_i = Minimum performance at which training device i may be use to train task in question. [$EPMIN_{Ti}$]

h_i = Hourly cost rate for training device i. [B_{in} where n represents current cost curve segment for device i; or $VHRC_i$]

i = Index of training device, usually the "incumbent" best device at current stage of allocation/assignment. [i]

j = Index of other, "candidate" training devices. [j]

k = Learning curve shape parameter. [r]

m_i = Time multiplier. [TM_{Ti}]

P_e = Entry performance level. [ENT_T]

S_i = Head start for training device i. (Same as S_d, earlier.) [HS_{Ti}]

t = Time each individual student spends in training on a particular device. [t]

T_i^- = Minimum allocation that may be made to device i for training current task under current cost curve constraints. [n.a.]

T_i^+ = Maximum allocation that may be made to device i for training current task under current cost curve constraints. [n.a.]

u_i = Effective upper bound on allocation to device i for current subproblem, considering both user-defined, cost curve segment-defined, or algorithm-imposed constraints. [UB_{Ti}]

U_i = User-defined upper bound on allocation to device i. [$USMAX_i$]

y = Performance level, in general. [n.a.]

y_j = Crossover performance level, at which dy/dD for TD_j just equals that of incumbent TD_i (at same performance level). [PX_{ij} or PX_{si}]

y_{now} = Current performance level. [P_{NOW}]

Y = Training standard for particular task. [STD_T]

References for Appendix B

Card, S. K., Moran, T. P., and Newell, A. (1983). The psychology of human-computer interaction. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.

Newell, A., and Rosenbloom, P. S. (1981). Mechanisms of skill acquisition and the law of practice. In J. R. Anderson (Ed.), Cognitive Skills and Their Acquisition (pp. 1-55). Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.

EXECUTIVE SUMMARY

This paper traces the history of overseas deployment training of the Army and Air National Guard and the intervention of a few governors in it beginning in 1985.

The furor after Governor Joseph Brennan of Maine canceled a deployment of two of his Army Guard units to Central America in 1986 prompted Congress to enact what has become known as the Montgomery Amendment, named after its author, U.S. Representative G.V. ``Sonny'' Montgomery (D-Mississippi). The amendment, which was added to the FY87 Department of Defense Authorization Act, withdraws from governors authority to withhold overseas deployments for their National Guard units on account of location, purpose, type or schedule of such training. A lawsuit brought by the governor of Minnesota seeks to have that language ruled unconstitutional as a violation of the Militia Clause to the U.S. Constitution.

The governors' actions had a major impact at the Department of Defense. They threatened the resources of the National Guard as it has evolved as a part of the Total Force. There was talk of withdrawing force structure from the Guard. In the end, the chief of the National Guard Bureau moved against the governor of Ohio, threatening to ``withdraw the Ohio National Guard from Ohio'' if Governor Richard Celeste didn't consent to the deployment of the 16th Engineer Brigade to Honduras in 1989. Faced with a $256 million-a-year bill, he did and it did.

# # #